VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

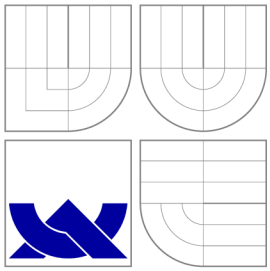
EXTRAKCE KLÍČOVÝCH SLOV Z VĚDECKÝCH ČLÁNKŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

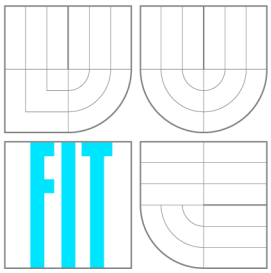
AUTOR PRÁCE
AUTHOR

Bc. MAREK KYJOVSKÝ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXTRAKCE KLÍČOVÝCH SLOV Z VĚDECKÝCH ČLÁNKŮ

KEYWORD EXTRACTION FROM SCIENTIFIC ARTICLES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MAREK KYJOVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2010

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2009/2010

Zadání diplomové práce

Řešitel: **Kyjovský Marek, Bc.**
Obor: Inteligentní systémy
Téma: **Extrakce klíčových slov z vědeckých článků**
Keyword Extraction from Scientific Articles
Kategorie: Umělá inteligence

Pokyny:

1. Seznamte se s metodami extrakce klíčových slov.
2. Shromážděte datovou sadu pro průběžné testování systému.
3. Navrhněte a implementujte systém pro automatickou extrakci klíčových slov z odborných publikací.
4. Vyhodnoťte vytvořený systém pomocí standardních metrik.

Literatura:

- podle dohody

Při obhajobě semestrální části diplomového projektu je požadováno:

- prototyp systému

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Smrž Pavel, doc. RNDr., Ph.D.**, UPGM FIT VUT
Datum zadání: 21. září 2009
Datum odevzdání: 26. května 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Hlavním cílem této práce je prozkoumat základní metody používající se k extrakci důležitých slov z článku. Poté se pokusit porozumět charakteru používaných klíčových slov z dostupné množiny testovacích anglických článků. Na základě těchto poznatků poté navrhnout a implementovat systém využívající tyto metody. Vytvořený systém pak otestovat na reálných anglických článcích a výsledky se pak pokusit vyhodnotit.

Abstract

The main goal of this thesis is to explore basic methods which is using for extraction of important words from articles. After that try to understand character of using keywords from the available set of testing English articles. Based on these findings, try to design and to implement a system which is using this methods. Then created system testing on the real English articles and after that try to analyse results.

Klíčová slova

klíčová slova, výrazy, vyhledávání klíčových slov, značkování, výběr výrazů, extrakce

Keywords

terms extraction, keywords, terms, index terms, keywords search, tagging, extraction

Citace

Marek Kyjovský: Extrakce klíčových slov z vědeckých článků, diplomová práce, Brno, FIT VUT v Brně, 2010

Extrakce klíčových slov z vědeckých článků

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. RNDr. Pavla Smrže, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Marek Kyjovský
25. května 2010

Poděkování

Na tomto místě bych rád poděkoval doc. RNDr. Pavlu Smržovi, Ph.D, za poskytnutí odborné pomoci, užitečných rad a za veškerý čas, který mi věnoval.

Obsah

Kapitola 1

Úvod

Vědecké články a publikace obsahují teoretické a praktické výsledky vědeckých výzkumů různých vědních oborů. Publikují se většinou v anglickém jazyce. Zdroje vědeckých článků mohou být různé. Nejčastěji se články publikují ve speciálních vědeckých časopisech. Dalšími důležitými zdroji jsou také různé akademické konference, odborné knihy či internet. Každý vědecký článek by měl pro jednodušší orientaci obsahovat seznam klíčových slov. Mnohdy to však není pravidlem. Časová náročnost ruční extrakce klíčových slov vede ke snaze vytvořit nástroj, který by klíčová slova extrahoval automaticky, nebo alespoň extrakci klíčových slov urychlil.

Klíčová slova (výrazy, termíny) se používají proto, aby si mohli čtenáři pouhým přečtením těchto slov udělat rychlý obrázek o obsahu článku. V ideálním případě se jedná o seznam základních, nejdůležitějších pojmů, které dokument dále osvětluje. Klíčovým slovem však často bývá také název tématu práce, jeho obor, nebo další důležité pojmy týkající se problematice daného článku. Klíčová slova mají především informační charakter, který může sloužit uživateli k vyhledávání určité publikace či článku. Další význam můžou mít například při automatickém třídění publikací podle tématu (v knihovně), nebo při tvorbě různých hierarchií publikací a článků.

Klíčová slova můžou být různě dlouhé. Obvykle se však setkáváme s jednoslovnými (unigramy), dvouslovnými (bigramy) a tříslovnými výrazy (trigramy). Každá publikace obvykle bývá označena více klíčovými slovy.

Správné určení klíčových slov automaticky není vůbec jednoduché. Většinou se jedná o subjektivní výběr autora. Slovo, které autor označí jako klíčové, nemusí nezávislému čtenáři připadat jako důležité. Naopak by mu jiný pojem nebo výraz mohl připadat důležitější. V některých případech se vhodné klíčové slovo v článku dokonce nemusí vůbec vyskytovat. V těchto případech je nemožné bez dalších podpůrných informacích o článku takové klíčové slovo automaticky určit.

Hlavním cílem této práce je prozkoumat základní metody používající se k extrakci důležitých slov z článku. Dále se pokusit porozumět charakteru používaných klíčových slov z dostupné množiny testovacích anglických článků. Poté se pokusit navrhnout a implementovat systém využívající tyto metody. Vytvořený systém pak otestovat na reálných anglických článcích a výsledky se pak pokusit vyhodnotit.

Text jsem rozdělil do devíti částí. První kapitolou, kterou právě čtete, je úvod. V další kapitole popisuji základní techniky určování vhodných slovních spojení. Třetí kapitola teoreticky popisuje některé statistické metody ohodnocování klíčových výrazů. Čtvrtá kapitola obsahuje popis a základní charakteristiku shromážděných testovacích dat. Pátá kapitola popisuje základní nástroje, které jsem při tvorbě a testování použil. Šestá kapitola popisuje

úvodní experimenty zabývající především porozumění charakteru typických klíčových výrazů. Problematiku návrhu a implementace systému popisuje sedmá kapitola. Najdete v ní také popis ovládání a strukturu používaných datových souboru. V osmé kapitole popisuji testy a výsledky, kterých jsem se svým systémem dosáhl. Poslední, devátou kapitolou je závěr.

Kapitola 2

Techniky určování vhodných slovních spojení

Při vyhledávání klíčových slov v článcích je nejprve potřeba si uvědomit, jaká slova hledat. Často jsou klíčová slova jednoslovná, také označovaná jako unigramy. Takové slova jsou většinou podstatná jména v 1. pádě jednotného čísla (základním tvaru). Jako klíčová slova jsou však často označovány i víceslovné výrazy. Nejčastěji dvouslovné a tříslovné výrazy (bigramy a trigramy). V těchto případech se mohou skládat také z přídavných jmen, případně sloves. Hlavním cílem této kapitoly bude popsat základní techniky určování vhodných slovních spojení.

2.1 Směrodatná odchylka

Výrazy se nemusí vždy skládat pouze ze sousedních slov. Někdy můžou být mezi slovy jednoho výrazu „díry“. Nejjednodušší technikou určení relevantnosti slovních spojení je určení směrodatné odchylky [7].

Nejprve je potřeba určit průměrnou vzdálenost slov ve větách. Ta je dána vztahem:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{2.1}$$

Směrodatná odchylka se poté určí ze vztahu:

$$\sigma^2 = \frac{\sum_{i=1}^{n} (d_i - \mu)^2}{n - 1}, \tag{2.2}$$

kde n je kolikrát se slova vyskytla společně, d_i je vzdálenost vzájemného výskytu vzhledem k pozici i a μ je průměrná vzdálenost výskytu.

Blíží-li se hodnota směrodatné odchylky nule, je pravděpodobné, že se jedná o kolokaci. V případě, že se odchylka rovná nule, pak se slova kolokace v textu vyskytují pouze vedle sebe.

2.2 Testování hypotéz

I přes vysokou četnost a nízkou standardní odchylku se slova spolu mohou vyskytovat pouze náhodou [7]. Aby se tato náhoda mohla vyvrátit, provádí se obvykle statistické testování hypotéz.

Nejprve stanovíme nulovou hypotézu H_0, která říká, že slova v n-gramu se vyskytují společně pouhou náhodou. K ní vytvoříme inverzní hypotézu H_1, která prozměnu říká, že slova jsou kolokací.

Následně vypočítáme pravděpodobnost p s jakou událost nastane, pokud je H_0 pravdivá. Pro slova vyskytující se spolu náhodně platí:

$$P(s_1 s_2) = P(s_1)P(s_2), \tag{2.3}$$

kde s_1 a s_2 jsou jednotlivá slova a P je pravděpodobnost. Pokud je pravděpodobnost hypotézy H_0 velmi malá, je možné hypotézu zavrhnout a tím potvrdit H_1.

Pravděpodobnosti se určují pomoci statistických testů, jako je např. t-test, nebo Pearsonův X^2 test.

2.3 T-test

T-test je metodou matematické statistiky, která umožňuje ověřit některý z následujících předpokladů [12]:

1. zda normální rozdělení, z něhož pochází určitý náhodný výběr, má určitou konkrétní střední hodnotu, přičemž rozptyl je neznámý
2. zda dvě normální rozdělení mající stejný rozptyl, z nichž pocházejí dva nezávislé náhodné výběry, mají stejné střední hodnoty

Hodnoty jednotlivých pravděpodobností hypotéz z předchozí kapitoly se dají určit právě t-testem, pomoci následujícího vzorce:

$$t = \frac{\overline{x} - \mu}{\sqrt{\frac{s^2}{N}}}, \tag{2.4}$$

kde $\overline{x}$ je střední hodnota vzorku, s^2 je odchylka vzorku, N je velikost vzorku a μ je střední hodnota celé množiny, ze které se vzorek vybírá. Jestliže je t dost velké, pak můžeme nulovou hypotézu zavrhnout. K hodnotám t se vyhledají stupně významnosti v statistických tabulkách. Jestliže t je větší než vyhledaný stupeň, pak můžeme nulovou hypotézu zavrhnout s pravděpodobností závislou na rozdílu t a stupně významnosti [7].

2.4 Pearsonův X^2 test

Pearsonův X^2 test je další metodou použitelnou při hledání relevantních slovních spojení. Je vhodnější než t-test, protože ten se zakládá na normálním pravděpodobnostním rozložení, což zcela neodpovídá povaze textových korpusů.

Pro X^2 test je vytvořena tabulka výskytů slovních spojení. Základ testu pak spočívá v porovnání frekvencí slovních spojení v tabulkách s frekvencí očekávanou pro nezávislá slova výrazu. Pokud je rozdíl mezi těmito frekvencemi příliš velký, pak je možné zavrhnout nulovou hypotézu. X^2 je asymptoticky rozložená χ^2, takže čím vyšší jsou hodnoty, tím větší je šance, že X^2 má rozložení χ^2 [7].

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \tag{2.5}$$

kde i udává počet řádků tabulky, j počet sloupců, O_{ij} je právě zkoumaná buňka tabulky a E_{ij} je očekávaná hodnota.

Důvodem proč se více používá X^2 test je to, že X^2 lze uplatnit i tam, kde je operováno s vysokými mírami pravděpodobnosti, kde by běžný t-test selhal.

2.5 Metody založené na gramatice

Dalším možným přístupem k vyhledávání relevantních spojení může být zpracování založené na lingvistice. Pro hledání slovních spojení je možné použít gramatické vzorce. Vyhledávání je založené na textu, ze kterého se automaticky vypíší všechny víceslovné pojmy odpovídající gramaticky a kolokačně. Kostry jsou hledány na základě gramatických a kolokačních vlastností. K zadanému textu se přidají gramatické vzorce platné pro jazyk, v jakém byl text vytvořen. Systém poté dokáže vytvořit seznam víceslovných výrazů [7].

Kromě frekvenčního vyhledávání jsou v kostrách použity gramatické vzorce. Spíše než frekvenční hledání klíčového slova jsou vyhledávány všechny relace, ve kterých se slovo vyskytuje. Slova jsou předzpracována morfologickým analyzátorem, který jim určí slovní druh a na výstupu jsou lemmatizována.

Kapitola 3

Metody ohodnocování klíčových slov

V této kapitole bych chtěl stručně shrnout základní statistické metody vyhledávání klíčových slov. Všechny tyto metody vycházejí ze statistických vlastností textu.

3.1 Četnost slov

Jedná se o nejjednodušší způsob vyhledávání klíčových slov [10]. Na základě četnosti slov v textu se určí klíčová slova. Tato metoda funguje dobře pro víceslovné výrazy (kolokace). K selhání metody však dochází při určování jednoslovných výrazů (unigramy), protože mezi nejčastěji vyskytující se slova v textech patří převážně předložky, spojky a zájmena.

V případě určování unigramů se metoda stává zajímavou až při použití morfologického analyzátoru [10]. Ten odfiltruje slova s nevhodnými slovními druhy. Poté se dá očekávat, že slova s vyšší frekvencí výskytu mohou být vhodnými kandidáty na klíčová slova.

Tato metoda je také nevhodná pro určení klíčových slov v krátkých textech, které nejsou dostatečně velkým vzorkem pro určení četnosti slov.

3.2 Backgroundový model

Background je seznam slov a jejich četnosti v obecném textu. Aby bylo určování klíčových slov pomoci tohoto modelu kvalitní, je potřeba, aby byl background tvořen z dostatečně velkého množství slov (miliony). Čím více obsahuje slov, tím by měl být kvalitnější [10].

Hlavním principem výběru klíčových slov pomoci backgroundového modelu je výběr výrazů, které se v backgroundu vyskytují velmi málo, nebo vůbec. Takové slova, nebo slovní spojení jsou tedy málo používané a s vysokou pravděpodobností specifické pro zpracovávaný text, což značí, že by se mohlo jednat o klíčové slovo.

Backgroundový model může být:

- Korpusový - background je vytvořen z obecných textů.
- Doménový - background je tvořen pouze texty z oblasti, o které zkoumaný text pojednává a snaží se vyhnout výběru slov, která jsou pro danou problematiku příliš obecná (např. *plane* v článku o typu letadla).

3.3 Term Frequency

Určením *Term Frequency (TF)* získáme frekvenci výskytu výrazu ve zpracovávaném dokumentu. Jedná se o jakousi normalizovanou četnost výrazu popsanou v 3.1. $Tf(i)$ vypočítáme podílem četnosti výrazu i v článku s celkovým počtem výrazů v článku.

$$Tf(i) = \frac{f(i)}{\Sigma_k f(k)}, \tag{3.1}$$

kde f_i je počet výskytů uvažovaného výrazu v textu a jmenovatel představuje počet všech výrazů v dokumentu.

3.4 Term Frequency - Inverse Document Frequency

Jedná se o často používaný algoritmus určování klíčových slov. Je založen na předpokladu, že výrazy s vyšší četností v jednom dokumentu mají v tomto dokumentu větší důležitost i přes jejich vzácný výskyt v celé kolekci dokumentů (background) [5].

Pro výpočet *tf-idf* je nejprve potřeba spočítat frekvenci výskytů výrazů v článku. To provedeme podle vzorce 3.1. Dalším krokem je vypočet *IDF*. $Idf(i)$ je míra důležitosti výrazu i v celé kolekci článků:

$$Idf(i) = \ln \frac{|D|}{\{d_j : t_i \in d_j\}}, \tag{3.2}$$

kde D je celkový počet dokumentů v kolekci. Jmenovatel je vyjádření počtu dokumentů obsahující zkoumaný výraz. Tento počet musí být vždy alespoň 1.

Hodnotu *Term Frequency - Inverse Document Frequency* dostaneme součinem $Tf(i)$ a $Idf(i)$:

$$Tfidf(i) = Tf(i) \cdot Idf(i). \tag{3.3}$$

Jako příklad můžeme mít v dokumentu 100 výrazů. Výraz *plane* se v textu vyskytuje 3 krát. V návaznosti na dříve definované vzorce, je tedy $Tf(i)$ pro výraz *plane* $(3/100) = 0,03$. Nyní předpokládejme, že máme 1 milion dokumentů a *plane* se objeví ve 100 z nich. Potom je $Idf(i) = ln(1\,000\,000/1\,00) = 9,21$. Celkový výsledek $Tfidf(i)$ pro výraz *plane* je tedy $0,03 \cdot 9,21 = 0,28$.

3.5 Residual IDF

Residual IDF je alternativní metodou k *IDF*. Metoda se snaží upřednostňovat termíny které nejsou náhodné. Přesněji řečeno *RIDF* je definována jako rozdíl mezi logaritmem skutečné frekvence termínu v dokumentu a frekvence termínu v dokumentu predikovanou Poissonovým rozdělením [5]:

$$RIDF(i) = Idf(i) - \log(1 - p(0; \lambda(i))), \tag{3.4}$$

kde p je Poissonovo rozdělení s parametrem $\lambda(i) = \frac{f(i)}{D}$ (průměrný počet výskytu termínu i v jednom dokumentu). f_i je celkový počet termínu i v celé kolekci dokumentů, D je pak celkový počet dokumentů v této kolekci. $1 - p(0; \lambda(i))$ je Poissonova pravděpodobnost dokumentu s méně než jedním výskytem termínu i.

Obecné Poissonovo rozdělení pravděpodobnosti lze vypočítat ze vzorce:

$$p(n, \lambda) = \frac{\lambda^n e^{-\lambda}}{n!}, \tag{3.5}$$

kde e je základ přirozeného logaritmu, n je počet výskytů určité události, a λ je kladné reálné číslo, rovnající se očekávanému počtu událostí, které mohou nastat.

Při výpočtu *RIDF* se tedy p vyčíslí následovně:

$$p(0, \lambda) = \frac{(\frac{f(i)}{D})^0 e^{-\frac{f(i)}{D}}}{0!} = e^{-\frac{f(i)}{D}}. \tag{3.6}$$

3.6 Weirdness

Weirdness je založeno na myšlence, že rozložení termínu ve specializovaném korpusu (doméně) je oproti obecnému korpusu významně odlišné. To můžeme vyjádřit následující formulí:

$$Weirdness(i) = \frac{\frac{f_s(i)}{n_s}}{\frac{f_g(i)}{n_g}}, \tag{3.7}$$

kde $f_s(i)$ a $f_g(i)$ jsou frekvence termínu i ve specializovaném, resp. obecném korpusu, n_s a n_g jsou celkové počty termínů v těchto korpusech [5].

Kapitola 4

Testovací data

Shromáždění kvalitních testovacích dat je základem pro budoucí tvorbu funkčního systému. Jelikož je mým cílem vytvořit systém pro automatický výběr klíčových slov, testovací data by měla obsahovat jakýsi rejstřík klíčových slov daného dokumentu.

Další důležitý požadavek na testovací data je jejich velikost. Čím více dat je k dispozici, tím lépe se výsledný systém vytváří a optimalizuje. Důležitá je také pestrost nashromážděných dat. To proto, aby se dala ověřit univerzálnost vytvořeného systému. Je lépe mít data z více zdrojů, než pouze z jednoho.

Podařilo se mi shromáždit celkem 11 074 článků s 39 440 klíčovými slovy. Tyto data pocházejí z 9-ti datových zdrojů – většinou se jedná o články z odborných konferencí. V dalších podkapitolách tyto zdroje podrobněji rozepíšu.

ASRU workshop

Jedná se o články z konference ASRU 2007 (IEEE Automatic Speech Recognition and Understanding Workshop) konaném v Kyotu (JAP). V každém článku jsou uvedeny „*Index Terms*“. Tyto klíčové výrazy jsem musel z článků automaticky vyextrahovat. Celkem se mi podařilo ze 101 článků vyextrahovat 395 klíčových slov. Každý článek tedy průměrně obsahuje 3,91 klíčových slov. Podrobné statistiky jsou zobrazeny v tabulce 4.1.

počet klíčových slov: 1	počet článků: 1
počet klíčových slov: 2	počet článků: 15
počet klíčových slov: 3	počet článků: 21
počet klíčových slov: 4	počet článků: 34
počet klíčových slov: 5	počet článků: 20
počet klíčových slov: 6	počet článků: 8
počet klíčových slov: 7	počet článků: 1
počet klíčových slov: 10	počet článků: 1

Tabulka 4.1: Statistika klíčových slov a článků konference ASRU 2007.

Konference ICASSP

Z konference ICASSP (IEEE International Conference on Acoustics, Speech and Signal Processing) mám články z roku 2007 (Honolulu) a 2008 (Las Vegas). Stejně jako u článku z ASRU jsou zde v textu článku uvedeny „*Index Terms*“. Tyto výrazy jsem musel automaticky vyextrahovat.

Konference mají celkem 2 427 článků (2007 – 1 188 článků, 2008 – 1 239 článků) obsahující klíčová slova. Vyextrahováno bylo 8 933 klíčových slov (2007 – 4 200 výrazů, 2008 – 4 633 výrazů). Průměrně tedy vychází 3,90 výrazů na článek. Podrobné statistiky v tabulce 4.2.

počet klíčových slov: 1	počet článků: 28
počet klíčových slov: 2	počet článků: 222
počet klíčových slov: 3	počet článků: 539
počet klíčových slov: 4	počet článků: 718
počet klíčových slov: 5	počet článků: 725
počet klíčových slov: 6	počet článků: 43
počet klíčových slov: 7	počet článků: 9
počet klíčových slov: 8	počet článků: 2
počet klíčových slov: 10	počet článků: 1

Tabulka 4.2: Statistika klíčových slov a článků konferencí ICASSP 2007 a ICASSP 2008.

Konference ICSLP

ICSLP – International Conference on Spoken Language Processing (Interspeech) se koná každoročně. Od roku 2006 jsou v každém článku definovány „*Index Terms*“. Opět bylo potřeba tyto klíčové výrazy z článku vyextrahovat.

Celkem mám staženy články 3 konferencí (2006, 2007, 2008). Bylo vyextrahováno 7 065 klíčových slov v 1 912 článcích (2006 – 589 článků, 2007 – 643 článků, 2008 – 680 článků). Průměrně to tedy dělá 3,70 výrazů na článek. Podrobné statistiky všech ICSLP konferencí naleznete v tabulce 4.3.

počet klíčových slov: 1	počet článků: 19
počet klíčových slov: 2	počet článků: 221
počet klíčových slov: 3	počet článků: 672
počet klíčových slov: 4	počet článků: 564
počet klíčových slov: 5	počet článků: 335
počet klíčových slov: 6	počet článků: 66
počet klíčových slov: 7	počet článků: 24
počet klíčových slov: 8	počet článků: 9
počet klíčových slov: 9	počet článků: 1
počet klíčových slov: 12	počet článků: 1

Tabulka 4.3: Statistika klíčových slov a článků konferencí ICSLP 2006, 2007 a 2008.

Konference EMBS

Konference EMBS (IEEE Engineering in Medicine and Biology Society) se v roce 2002 konala v Janově. Bylo publikováno 1 091 článků obsahujících „*Keywords*“. Opět bylo potřeba klíčová slova z článků automaticky vyextrahovat.

Celkem bylo v článcích nalezeno 4 213 klíčových slov. Průměrně tedy 3,86 výrazů na článek. Podrobnější statistiky najdete v tabulce 4.4.

počet klíčových slov: 1	počet článků: 3
počet klíčových slov: 2	počet článků: 88
počet klíčových slov: 3	počet článků: 393
počet klíčových slov: 4	počet článků: 331
počet klíčových slov: 5	počet článků: 167
počet klíčových slov: 6	počet článků: 80
počet klíčových slov: 7	počet článků: 19
počet klíčových slov: 8	počet článků: 7
počet klíčových slov: 9	počet článků: 3

Tabulka 4.4: Statistika klíčových slov a článků konference EMBS.

Konference Coling

Jedná se o konferenci Coling 2000 (International Conference on Computational Linguistics) konanou v Saarbrückenu (GER). Klíčová slova jsou uložena v metadatech. Ty obsahují podpůrné informace ke každému článku konference (autory, abstrakt, reference atd.). Mám staženy i následující konference Coling 2002 a Coling 2004. Bohužel u těchto konferencí jsou klíčová slova v metadatech nepoužitelná.

Coling 2000 obsahuje 104 článků s 490 klíčovými slovy. Průměrně tedy na jeden článek vychází 4,71 klíčových slov. Statistiky klíčových slov jsou popsány v tabulce 4.5.

počet klíčových slov: 1	počet článků: 2
počet klíčových slov: 2	počet článků: 8
počet klíčových slov: 3	počet článků: 22
počet klíčových slov: 4	počet článků: 26
počet klíčových slov: 5	počet článků: 17
počet klíčových slov: 6	počet článků: 9
počet klíčových slov: 7	počet článků: 10
počet klíčových slov: 8	počet článků: 4
počet klíčových slov: 9	počet článků: 3
počet klíčových slov: 10	počet článků: 2
počet klíčových slov: 14	počet článků: 1

Tabulka 4.5: Statistika klíčových slov a článků konference Coling 2000.

Konference LREC

LREC – The International Conference on Language Resources and Evaluation se koná od roku 1998 každé 2 roky. Klíčová slova každého článku konference jsou uloženy v metadatech obsahujících další podpůrné informace.

Mám staženy konference LREC 2000 (204 článků), LREC 2002 (321 článků) a LREC 2004 (495 článků) – dohromady 1 020 článků. Celkem se v metadatech nachází 3 829 klíčových slov. Průměrně to tedy na článek dělá 3,75 klíčových slov. Podrobnou statistiku naleznete v tabulce 4.6.

počet klíčových slov: 1	počet článků: 125
počet klíčových slov: 2	počet článků: 101
počet klíčových slov: 3	počet článků: 232
počet klíčových slov: 4	počet článků: 214
počet klíčových slov: 5	počet článků: 239
počet klíčových slov: 6	počet článků: 59
počet klíčových slov: 7	počet článků: 25
počet klíčových slov: 8	počet článků: 13
počet klíčových slov: 9	počet článků: 6
počet klíčových slov: 10	počet článků: 3
počet klíčových slov: 11	počet článků: 1
počet klíčových slov: 13	počet článků: 1
počet klíčových slov: 14	počet článků: 1

Tabulka 4.6: Statistika klíčových slov a článků konferencí LREC 2000, 2002 a 2004.

Konference GWN

První konference GWN (International Global WordNet Conference) proběhla v roce 2002 v Mysore (IND). Konference probíhá každé 2 roky. Mám staženy články konference GWN 2002. Klíčová slova k článkům jsou uložena v metadatech. Bohužel je článků se zadanými klíčovými slovy relativně málo – celkem 30. V těchto článcích se nachází 116 klíčových slov, což je v průměru 3,87 výrazu na článek. Statistiky ke konferenci GWN 2002 naleznete v tabulce 4.7.

počet klíčových slov: 1	počet článků: 2
počet klíčových slov: 2	počet článků: 4
počet klíčových slov: 3	počet článků: 5
počet klíčových slov: 4	počet článků: 9
počet klíčových slov: 5	počet článků: 7
počet klíčových slov: 6	počet článků: 2
počet klíčových slov: 8	počet článků: 1

Tabulka 4.7: Statistika klíčových slov a článků konference GWN 2002.

Oxford Journals

Oxford University Press je největší univerzitní vydavatelství na světě. Vydává spoustu předních světových vědeckých časopisů, které jsou rozčleněny do 6-ti kategorii [1]:

- Humanities
- Law
- Life Sciences
- Mathematics & Physical Sciences
- Medicine
- Social Sciences

Časopisy a jejich články je možno stahovat na serveru oxfordjournals.com. Podařilo se mi automaticky stáhnout 2 702 článků. Tyto články bohužel nemám rozčleněny do kategorií. Všechny však obsahují klíčová slova v metadatech. Ve všech stažených článcích je celkem 8 723 klíčových slov. Průměrně tedy 3,23 klíčových slov na jeden článek. Podrobné statistiky naleznete v tabulce 4.8. Oproti konferencím popsanými výše, obsahuje tento zdroj poměrně mnoho článků s jedním nebo dvěma klíčovými výrazy, což je docela málo.

počet klíčových slov: 1	počet článků: 463
počet klíčových slov: 2	počet článků: 652
počet klíčových slov: 3	počet článků: 609
počet klíčových slov: 4	počet článků: 441
počet klíčových slov: 5	počet článků: 247
počet klíčových slov: 6	počet článků: 146
počet klíčových slov: 7	počet článků: 68
počet klíčových slov: 8	počet článků: 28
počet klíčových slov: 9	počet článků: 20
počet klíčových slov: 10	počet článků: 8
počet klíčových slov: 11	počet článků: 6
počet klíčových slov: 12	počet článků: 7
počet klíčových slov: 15	počet článků: 3
počet klíčových slov: 18	počet článků: 1
počet klíčových slov: 22	počet článků: 2
počet klíčových slov: 38	počet článků: 1

Tabulka 4.8: Statistika klíčových slov a článků Oxford journals.

Databáze PNAS

PNAS – Proceedings of the National Academy of Sciences je oficiálním časopisem Národní akademie věd Spojených států amerických. Je jedním z nejcitovanějších multidisciplinárních vědeckých periodik. Časopis byl založen v roce 1914, přináší články, zprávy, komentáře a přehledy z řady vědeckých oborů. Vychází v tištěné podobě, ale také online na pnas.org [2].

Podařilo se mi automaticky stáhnout celkem 1 827 článků. Klíčová slova jsou uloženy v metadatech. Celkem jich je dostupných 5 934. Na jeden článek je to průměrně 3,25. Podrobné statistiky jsou vypsány v tabulce 4.9. Stejně jako u Oxford Journals obsahuje tento zdroj mnoho článků s poměrně nízkým počtem klíčových výrazů.

počet klíčových slov: 1	počet článků: 293
počet klíčových slov: 2	počet článků: 473
počet klíčových slov: 3	počet článků: 397
počet klíčových slov: 4	počet článků: 284
počet klíčových slov: 5	počet článků: 189
počet klíčových slov: 6	počet článků: 79
počet klíčových slov: 7	počet článků: 48
počet klíčových slov: 8	počet článků: 28
počet klíčových slov: 9	počet článků: 11
počet klíčových slov: 10	počet článků: 14
počet klíčových slov: 11	počet článků: 2
počet klíčových slov: 12	počet článků: 2
počet klíčových slov: 13	počet článků: 2
počet klíčových slov: 14	počet článků: 2
počet klíčových slov: 15	počet článků: 1
počet klíčových slov: 16	počet článků: 1
počet klíčových slov: 19	počet článků: 1

Tabulka 4.9: Statistika klíčových slov a článků Pnas.

4.1 Shrnutí

Celkem se mi podařilo nashromáždit 11 074 článků s 39 440 klíčovými slovy. Na jeden článek průměrně vychází 3,56 výrazů. Články většinou pochází z technických konferencí, najdeme však i články pocházející z jiných vědeckých oborů. Jde především o databáze PNAS a Oxford Journals. Podrobné statistiky o počtu výrazů v článcích najdete v tabulce 4.10. Statistiky jsou také zobrazeny v grafu 4.1.

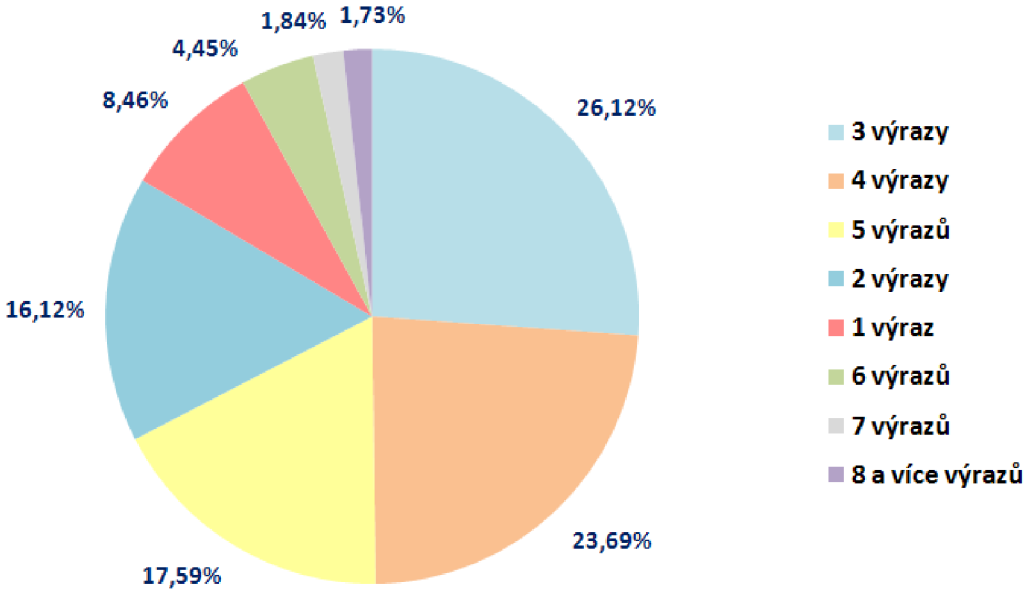
Zhruba u poloviny článků jsem měl k dispozici metadata, ve kterých již byla klíčová slova identifikována. Druhou polovinu však tvořily články, které tato metadata neměly. Přímo v textu těchto článků však byly definovány „*Index Terms*“, nebo „*Keywords*“, které označovaly právě klíčové výrazy v článcích. Pomoci vytvořeného skriptu jsem se pokusil tyto klíčové termíny vyextrahovat, což se mi relativně podařilo.

Z tabulky 4.10 a grafu 4.1 můžeme vyčíst, že nejčastěji se k článkům přiřazují 3-4 klíčová slova a to v polovině případů. Často se k vyjádření používá také 2 nebo 5 výrazů. Běžně se ke každému článku přidává maximálně do 6 klíčových výrazů. Jiné hodnoty nejsou příliš časté. V několika extrémních případech je však k článků přiřazeno dokonce více než 15 výrazů.

Pro jednodušší práci s testovacími daty jsem musel sjednotit jejich formát. Vytvořil jsem si XML soubor s jednoduchou strukturou pro snadný přístup ke všem článkům a jejich klíčovým výrazům. Příklad XML souboru můžete vidět na obrázku 4.2.

počet klíčových slov: 1	počet článků: 936
počet klíčových slov: 2	počet článků: 1784
počet klíčových slov: 3	počet článků: 2890
počet klíčových slov: 4	počet článků: 2621
počet klíčových slov: 5	počet článků: 1946
počet klíčových slov: 6	počet článků: 492
počet klíčových slov: 7	počet článků: 204
počet klíčových slov: 8	počet článků: 92
počet klíčových slov: 9	počet článků: 44
počet klíčových slov: 10	počet článků: 29
počet klíčových slov: 11	počet článků: 9
počet klíčových slov: 12	počet článků: 10
počet klíčových slov: 13	počet článků: 3
počet klíčových slov: 14	počet článků: 4
počet klíčových slov: 15	počet článků: 4
počet klíčových slov: 16	počet článků: 1
počet klíčových slov: 18	počet článků: 1
počet klíčových slov: 19	počet článků: 1
počet klíčových slov: 22	počet článků: 2
počet klíčových slov: 38	počet článků: 1

Tabulka 4.10: Statistika všech klíčových slov a článků.



Obrázek 4.1: Graf znázorňující počet článků s počtem přiřazených výrazů.

```
<clanky>
  <clanek>
    <nazev>nazev.txt</nazev>
    <konference>konference</konference>
    <tagy>
      <tag>klicovy_vyraz_1</tag>
      <tag>klicovy_vyraz_2</tag>
      <tag>klicovy_vyraz_3</tag>
    </tagy>
  </clanek>
  <clanek>
    <nazev>nazev2.txt</nazev>
    <konference>konference2</konference>
    <tagy>
      <tag>klicovy_vyraz_4</tag>
      <tag>klicovy_vyraz_5</tag>
      <tag>klicovy_vyraz_6</tag>
    </tagy>
  </clanek>

</clanky>
```

Obrázek 4.2: Struktura XML souboru s testovacími daty.

Tag `<nazev>` je název souboru s článkem a `<konference>` udává název konference, nebo datového zdroje článku. Klíčová slova přiřazené k článku se jsou uloženy v tagu `<tag>`. Předpokládá se, že samotný text článku najdeme ve složce `konference/txt/nazev.txt`.

Kapitola 5

Použité nástroje

Tato kapitola stručně popisuje nejdůležitější nástroje, které jsem při řešení použil. Jedná se o programovací jazyk Python a další nástroje vyvíjené převážně na naší fakultě.

5.1 Python

Python je dynamický objektově orientovaný programovací jazyk, který v roce 1990 navrhl Guido van Rossum. Python je vyvíjen jako open source projekt, který zdarma nabízí instalační balíky pro většinu běžných platforem (Unix, Windows, Mac OS); ve většině distribucí systému Linux je Python součástí základní instalace [11].

Někdy bývá Python zařazován mezi takzvané skriptovací jazyky. Jeho možnosti jsou ale větší. Python byl navržen tak, aby umožňoval tvorbu rozsáhlých, plnohodnotných aplikací (včetně grafického uživatelského rozhraní). Při psaní programů umožňuje Python používat nejen objektově orientované paradigma, ale i procedurální a v omezené míře i funkcionální, podle toho komu co vyhovuje nebo se pro danou úlohu hodí nejlépe. Python má díky tomu vynikající vyjadřovací schopnosti. Kód programu je ve srovnání s jinými jazyky krátký a dobře čitelný. Význačnou vlastností jazyka Python je produktivnost z hlediska rychlosti psaní programů [11].

ConvertPdf

Tento nástroj slouží k převodu článků z formátu PDF do textové podoby. Je vyvíjený na naší fakultě. Nástroj používá na samotný převod program pdftotext a pdfinfo, které jsou součástí programové balíku Xpdf [6].

Důležitou a užitečnou vlastností je hromadný převod několika dokumentů najednou. To uživateli ušetří spoustu času. Další zajímavou funkcí je zhodnocení správnosti převodu. Někdy se totiž stává, že se nepovede dokument správně převést. Na posouzení správnosti převodu se používají jednoduché heuristiky, založené na procentuálním výskytu nepovolených znaků a průměrném počtu slov na stranu. Za povolené znaky se považují jen znaky anglické abecedy, číslice, interpunkce a bílé znaky [6]. Jiné nástroje většinou neobsahuji funkci pro odhalení chybného převodu dokumentů.

5.2 TreeTagger

TreeTagger je nástroj sloužící ke značkování textů. Ke každému slovu v textu přiřadí informace o slovním druhu a základním tvaru slova (lemma). Tento nástroj byl vytvořen Helmutem Schmidem na univerzitě ve Stuttgartu. TreeTagger je úspěšně využíván ke značkování německých, anglických, italských, španělských, bulharských, ruských, řeckých, portugalských, čínských a francouzských textů a může být také adaptován pro další jazyky s dostupným slovníkem a manuálně značkovaným trénovacím korpusem [9].

Ve svém projektu používám obálku pro Python vytvořenou na naší fakultě, která vstupní text rozdělí na jednotlivá slova a postupně vrací pro každé slovo výsledek, obsahující původní slovo, jeho slovní druh a základní tvar. Slova v klíčových výrazech se většinou vyskytují v základních tvarech, proto je některých případech vhodné použít právě TreeTagger k jejich převodu do základních tvarů. Další využití má TreeTagger při určování slovních druhů a následné filtraci nevhodných druhů.

5.3 Related docs

Tento nástroj slouží k zjišťování sémantické blízkosti mezi dokumenty a je také vyvíjen na naší fakultě. Nástroj nejprve zaindexuje všechny dokumenty pomoci Solr. Z indexů se poté vytvoří matice podobností (sémantické blízkosti) mezi dokumenty, jejíž hodnoty vznikají skalárním součinem jednotlivých dokumentových vektorů. Každý řádek a sloupec matice podobnosti přestavuje hodnoty pro právě jeden dokument. Na diagonále se nacházejí kontrolní hodnoty, které by měli být vždy 1. Následně se hledají nejvyšší hodnoty, které reprezentují nejpodobnější dokumenty [4].

K indexaci je použit Apache Solr. Jedná se o populární a výkonný open source vyhledávací server založený na Lucene Java search library. Mezí hlavní vlastnosti patří rychlé fulltextové vyhledávání, dynamické shlukování, databázová integrace a bohaté možnosti manipulace s různými typy dokumentu (Word, PDF, atd.). Solr je vysoce škálovatelný, poskytuje distribuované vyhledávání a indexaci dokumentů. Je používaný také mnoha velkými světovými internetovými servery k vyhledávání a navigaci. Solr využívá Lucene Java search knihovnu a je jednoduché ho požívat prakticky z libovolného programovacího jazyka [3].

Kapitola 6

Úvodní experimenty

V této kapitole jsou popsány některé experimenty a testy, které jsem se pokusil vykonat nad testovacími daty. Jedná se převážně o zkoumání charakteristiky typicky používaných klíčových výrazů. Výsledky těchto experimentů můžou být důležité z hlediska další tvorby a optimalizace výsledného systému. Ve všech experimentech jsem použil jednoduché statistické metody. V jednom testu byla potřeba použít morfologický analyzátor TreeTagger k určení slovních druhů.

6.1 Délka slov

Tento test byl vytvořen za účelem zjistit, jak dlouhá slova se v klíčových výrazech nejčastěji vyskytují. To může být zajímavé vědět, z důvodu budoucí filtrace krátkých a dlouhých slov. Test jsem provedl nad množinou všech testovacích klíčových výrazů. Výsledky můžete vidět v tabulce 6.1. V prvním sloupci najdeme délku slova (počet znaků). Druhý sloupec obsahuje počet takových slov, v posledním sloupci je tento údaj znázorněn procentuálně ke všem slovům.

délka slova	počet slov	%
6 znaků	9 459 slov	14,64 %
8 znaků	7 812 slov	12,09 %
7 znaků	6 795 slov	10,51 %
5 znaků	6 647 slov	10,28 %
10 znaků	6 333 slov	9,80 %
9 znaků	6 249 slov	9,67 %
4 znaků	4 728 slov	7,31 %
11 znaků	4 661 slov	7,21 %
3 znaků	3 233 slov	5,00 %
12 znaků	2 565 slov	3,97 %
13 znaků	1 734 slov	2,68 %
14 znaků	1 431 slov	2,21 %
2 znaků	942 slov	1,45 %
ostatní		méně než 1 %

Tabulka 6.1: Délka slov

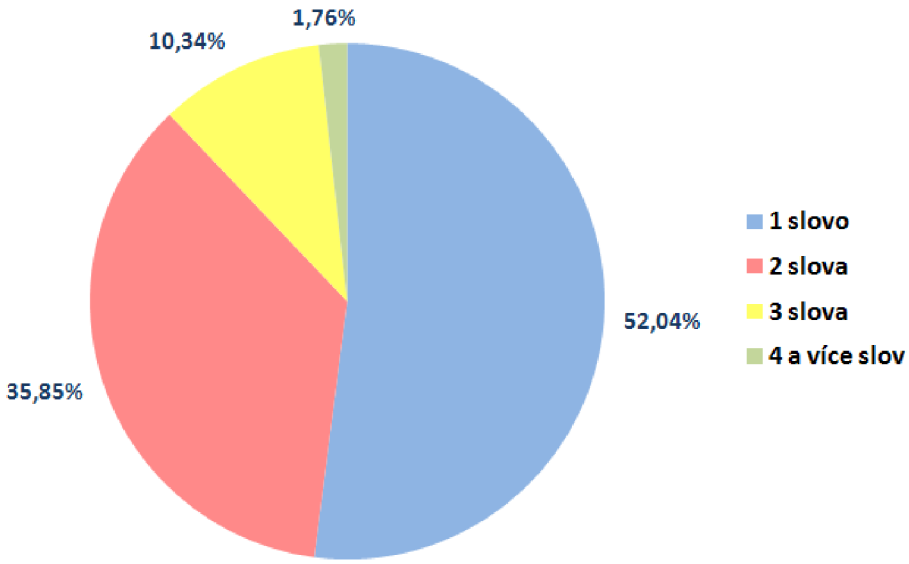
Z tabulky můžeme vyčíst, že nejčastější délky slov používaných v klíčovém výrazu je 5-8 znaků. Naopak je zřejmě zbytečné vyhledávat slova skládající se z jednoho, nebo dvou znaků. 3-znaková slova možná bude vhodné také odfiltrovat, jelikož se jich v klíčových výrazech nenachází mnoho, ale textech se vyskytují často. Odfiltrováním těchto slov docílíme výrazného úbytku kandidátních termínů, což by mohlo být výhodné z hlediska rychlosti zpracování, redukce potřebného paměťového prostoru a hlavně kvalitnějšího ohodnocení zbylých kandidátních termínů.

6.2 Délka výrazů

V tomto experimentu jsem zjišťoval délku klíčových výrazů (počet slov). Cílem bylo zjistit, jak dlouhé výrazy se v praxi nejčastěji používají. Výsledky experimentu jsou zadány v tabulce 6.2. První sloupec je délka výrazů. Ve druhém sloupci najdeme počet takových výrazů a v posledním sloupci je poměr ke všem výrazům zadaný v procentech. Grafické vyhodnocení experimentu najdete v grafu 6.1.

délka	počet výrazu	%
1 slovo	20 661 výrazů	52,04 %
2 slova	14 233 výrazů	35,85 %
3 slova	4 106 výrazů	10,34 %
4 slova	616 výrazů	1,55 %
5 slov	66 výrazů	0,17 %
6 a více slov	17 výrazů	0,04 %

Tabulka 6.2: Statistika délky klíčových výrazů



Obrázek 6.1: Graf znázorňuje délku klíčových výrazů.

Z výsledků vyplývá, že více než polovina výrazů (52,04 %) jsou unigramy. Nezanedbatelné jsou však také bigramy a trigramy (dohromady 46 %). Delší výrazy (1,76 %) nejsou v praxi příliš časté. V návrhu systému se tedy zaměřím pouze na výrazy délky 1-3 slov. Delší výrazy budu moci ignorovat.

6.3 Unikátnost výrazů

V tomto testu jsem se pokusil zjistit, kolik klíčových výrazů je jedinečných a jaké výrazy jsou nejčastější. Test jsem provedl jak pro jednotlivé konference zvlášť, tak také pro všechny výrazy společně. Hledal jsem také nejčastější klíčové výrazy pro každou konferenci. Výsledky najdete v tabulce 6.3.

konference	celkem výrazu	unikátních	%	nejčastější výraz
ASRU	395	281	71,13 %	speech recognition (26×)
ICASSP	8 933	3 770	42,20 %	speech recognition (138×)
ICSLP	7 065	3 085	43,66 %	speech recognition (267×)
EMBS	4 213	2 390	56,73 %	tissue engineering (40×)
Coling	490	381	77,75 %	machine translation (5×)
LREC	3 829	1 881	49,12 %	evaluation (86×)
GWN	116	83	71,55 %	wordnet (14×)
Oxford Journals	8 723	2 713	31,09 %	evolution (120×)
PNAS	5 934	2 375	40,02 %	evolution (86×)
CELKEM	39 440	13 710	34,53 %	speech recognition (456×)

Tabulka 6.3: Unikátnost výrazů

Nejčastějším klíčovým výrazem je „*speech recognition*“. Je tedy patrné, že většina konferencí se zabývá rozpoznávání řeči a přirozeného jazyka. Výjimku tvoří vědecké časopisy Oxford Journals a PNAS, jejichž nejčastějším klíčovým výrazem je „*evolution*“, což může znamenat, že většina článků se zabývá biologií.

Celkem 34,53% všech výrazů jsou unikátní. Znamená to, že se nevyskytují jako klíčový výraz v žádném jiném článku. Pokud se podíváme na výsledky jednotlivých datových zdrojů zvlášť, vidíme, že u některých konferencí je více než 70% všech výrazů unikátních. Naopak u článků Oxford Journals je to pouze 31%. Může to znamenat, že hodně článků se zabývá podobným tématem a klíčová slova se často opakují.

6.4 Výskyty výrazů v článku

V tomto experimentu jsem se pokusil k článku přiřazené klíčové výrazy vyhledat přímo v jejich textu. Článek totiž v určitých případech nemusí vůbec klíčový výraz obsahovat. Klíčovým výrazem může být např. nějaký obecnější termín, nebo název podoblasti. Je jasné, že takové termíny nemusí být v článku obsaženy vůbec. Bez znalostí dalších podpůrných informací o článku je pak téměř nemožné takový klíčový výraz automaticky identifikovat.

Výsledky experimentu jsem umístil do tabulky 6.4. První sloupec obsahuje počet výskytů klíčového výrazu v článku. Ve druhém sloupci tabulky najdeme počet výrazů odpovídající výskytům. Poslední sloupec udává poměr ke všem výrazům v procentech.

počet výskytů	počet výrazu	%
0 výskytů	11 699 výrazů	23,30 %
1 výskytů	3 778 výrazů	9,49 %
2 výskytů	2 898 výrazů	7,30 %
3 výskytů	2 375 výrazů	5,98 %
4 výskytů	1 892 výrazů	4,77 %
5 výskytů	1 744 výrazů	4,39 %
6 výskytů	1 478 výrazů	3,72 %
7 výskytů	1 275 výrazů	3,21 %
8 výskytů	1 091 výrazů	2,75 %
9 výskytů	998 výrazů	2,51 %
10 výskytů	845 výrazů	2,13 %
11-15 výskytů	3 151 výrazů	7,94 %
16-20 výskytů	2 047 výrazů	5,16 %
21-30 výskytů	2 452 výrazů	6,18 %
31-40 výskytů	1 306 výrazů	3,29 %
41-50 výskytů	834 výrazů	2,10 %
50 a více výskytů	2 296 výrazů	5,78 %

Tabulka 6.4: Statistika výskytů výrazů v článku

Z tabulky můžeme vyčíst, že překvapivě hodně klíčových výrazů (23,30 %) nenajdeme v přiřazeném článku vůbec. Je však problém tyto výrazy analyzovat automaticky. Pokusil jsem se tedy o „ruční“rozbor.

Ze vzorku dat, který jsem prošel ručně vyplynulo několik problémových skupin výrazů:

- tvar výrazu – Klíčový výraz je v jiném jazykovém tvaru než se vyskytuje v textu (množné číslo, čas atd.). To by mohlo být řešitelné TreeTaggerem a převodem výrazů do základního tvaru.
- obecnější termín – Klíčový termín je obecnějším výrazem pod který článek spadá, v textu článku se však tento výraz nevyskytuje.
- název konference – Klíčový termín je název konference pod který článek spadá, v textu článku se však nevyskytuje.
- zkratka – Klíčový výraz je zkratka některého termínu, který se textu vyskytuje. Nebo naopak klíčový termín je víceslovný výraz a v textu se pak používá pouze jeho zkratka.
- závorky – V některých klíčových výrazech jsem narazil také na závorky, které obsahují zkratku, nebo bližší specifikaci výrazu. V textu však najdeme pouze část takového výrazu – obsah závorky, nebo výraz před závorkou.
- chyby – Chybně extrahované, nebo chybně zadané klíčové termíny

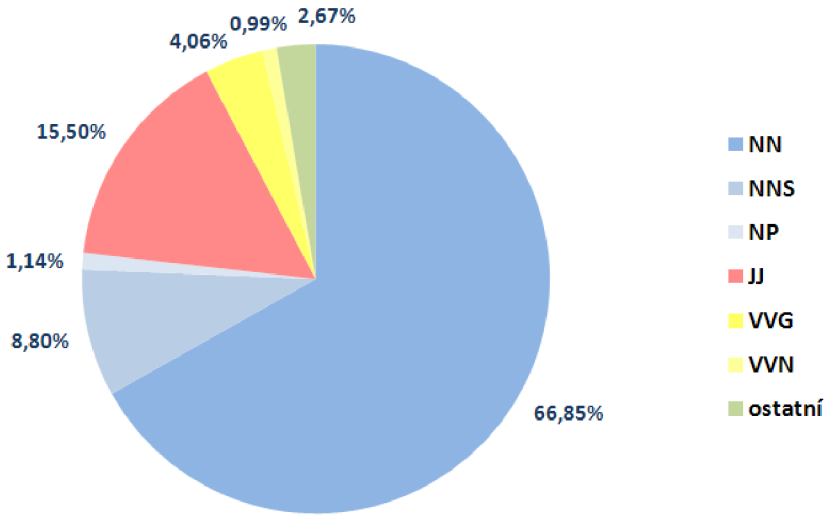
6.5 Slovní druhy klíčových výrazů

V tomto experimentu jsem se pokusil zjistit, ze kterých slovních druhů se klíčové výrazy nejčastěji skládají. Bylo potřeba použít morfologický analyzátor TreeTagger, který dokáže slova označkovat slovními druhy. Výsledky tohoto testu se dají použít k odfiltrování slov, jejichž slovní druh se v klíčových výrazech příliš nepoužívá.

V první části tohoto experimentu jsem zjišťoval slovní druhy jednotlivých slov. Výsledek můžeme vidět v tabulce 6.5. První sloupec udává slovní druh (vysvětlivky v A.1). Druhý sloupec udává počet slov s tímto druhem a v posledním sloupci najdeme vyjádření výsledku v procentech vzhledem ke všem slovům. Výsledky toho experimentu jsem se také pokusil znázornit v grafu 6.2.

druh slova	počet slov	%
NN	45 003	65,77 %
JJ	10 436	15,25 %
NNS	5 923	8,65 %
VVG	2 730	3,99 %
NP	767	1,12 %
VVN	664	0,97 %
VVZ	456	0,66 %
VVP	444	0,64 %
IN	331	0,48 %
VVD	316	0,46 %
VV	196	0,28 %

Tabulka 6.5: Druhy slov klíčových výrazů



Obrázek 6.2: Graf znázorňuje statistiku nejpoužívanějších klíčových výrazů.

Jasně nejpoužívanějším slovním druhem mezi klíčovými výrazy je podstatné jméno, které se vyskytuje v 66 % případů v jednotném čísle a téměř v 9 % případů v množném čísle. Celkem se tedy podstatné jméno vyskytuje zhruba v 75 % všech případů. Druhým nejpoužívanějším slovním druhem je přídavné jméno (15,25 %). Docela používaným slovním druhem je také sloveso v různých tvarech. Nejčastějším tvarem je gerundium. Posledním druhem, který stojí za zmínku jsou vlastní názvy a jména (1,12 %). Další slovní druhy se v klíčových výrazech příliš nepoužívají.

V druhé části tohoto experimentu jsem zjišťoval nejčastějším stavby klíčových výrazů podle slovních druhů. Zabýval jsem se pouze výrazy skládající se z jednoho, dvou, nebo tří slov. Výsledky nejčastějších typů můžeme najít v tabulce 6.6. Jednotlivé slovní druhy každého výrazu jsou v tabulce od sebe odděleny mezerou. Vysvětlení jednotlivých zkratek druhů slov najdete v A.1.

druh spojení	počet výrazů	%
NN	16 112	38,47 %
NN NN	6 948	16,59 %
JJ NN	3 475	8,29 %
NNS	3 299	7,87 %
JJ NN NN	1 586	3,78 %
JJ NNS	1 379	3,29 %
JJ	1 305	3,11 %
NN NNS	1 121	2,67 %
NN NN NN	833	1,98 %
NP	574	1,37 %
NN VVG	424	1,01 %
VVG	306	0,73 %
JJ JJ NN	236	0,56 %
JJ NN NNS	230	0,54 %
VVN NN	180	0,42 %
JJ VVG	173	0,41 %
VVN NN NN	172	0,41 %
NNS NN	154	0,36 %
VVG NN	139	0,33 %
NN NN NNS	123	0,29 %

Tabulka 6.6: Klíčové výrazy podle druhů

Stejně jako v první části tohoto experimentu, je také nejpoužívanějším klíčovým výrazem podstatné jméno (38,47 % jednotné číslo a 7,87 % množné číslo). Hojně používanými výrazy jsou také spojení dvou, nebo tří podstatných jmen za sebou. Nejpoužívanější tříčlenný výraz však začíná přídavným jménem, které následují dvě podstatná jména. Přídavné jméno se často vyskytuje na začátcích výrazů většinou ve spojení s podstatnými jmény. Často se vyskytují také vlastní názvy. V některých výrazech najdeme také sloveso v *-ing* tvaru (gerundium).

Kapitola 7

Návrh a implementace systému

V této kapitole se pokusím popsat návrh systému, jeho strukturu a následnou implementaci. Podrobněji zde popíšu jednotlivé kroky zpracování vstupního textu od načtení po určení nejdůležitějších klíčových slov. Popíšu zde také princip funkčnosti vytvořených skriptů. Poslední část jsem věnoval popisu ovládání těchto skriptů. Specifikuji zde potřebnou strukturu a formát vstupních a výstupních souborů.

7.1 Požadavky na systém

Hlavním cílem mého systému by mělo být nalezení klíčových slov ve vstupních článcích. Klíčová slova se používají hlavně ke stručnému popisu tématu daného článku. Díky nim může čtenář rychle a jednoduše vyhledávat články s požadovaným obsahem ve velké množině různých nesouvisejících publikací.

Problém hromadného zpracovávání článků spočívá hlavně v efektivitě vytvořeného systému. Čím bude systém složitější, tím by měl být výsledek zpracování kvalitnější. Bohužel však bude zpracování pomalejší. Proto je potřeba zvážit, které části systému budou opravdu přínosné z hlediska výpočetní složitosti.

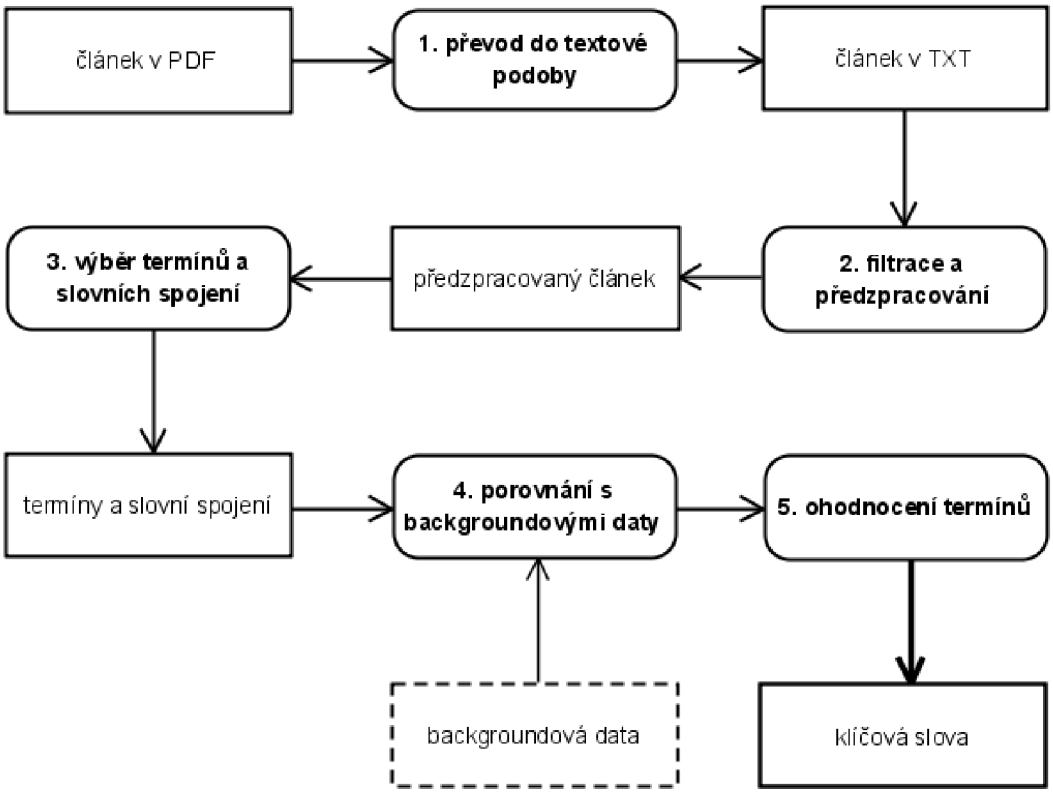
Další důležitou vlastností systému by měla být jeho nezávislost na vstupním formátu. Většina vědeckých článků je však ve formátu PDF. Zaměřil jsem se tedy pouze na tento formát.

7.2 Struktura systému

Obecně by se měl systém skládat z několika po sobě jdoucích kroků, které na sebe navazují:

1. převod článku z PDF do textové podoby
2. filtrace a předzpracování textu
3. výběr kandidátních termínu a slovních spojení
4. porovnání vybraných termínů s backgroundovým korpusem
5. ohodnocení termínů pomoci některé metody

Strukturu navrženého systému jsem se pokusil zobrazit na obrázku 7.1. V další části kapitoly jednotlivé kroky podrobněji popíšu.



Obrázek 7.1: Struktura navrženého systému

7.2.1 Převod článku z PDF do textové podoby

Vědecké články se můžou vyskytovat v různých formátech. Většina publikovaných článků se však nachází ve formátu PDF. V dokumentu tohoto typu jsou uloženy kromě samotného textu článku také další informace (formátování, komprese atd.).

V prvním kroku zpracování je tedy potřeba článek kvalitně převést z tohoto formátu do čistě textové podoby. Jelikož se jedná o docela složitou problematiku, rozhodl jsem se k tomuto účelu použít nástroj *convertPdf*, který jsem podrobněji popsal v kapitole 5.1. Pokud již je původní článek v textové podobě, tento krok se samozřejmě může vynechat.

7.2.2 Předzpracování textu

Druhým krokem je předzpracování a filtrace textu. Převod z předchozího kroku může obsahovat menší chyby a nepřesnosti, špatné převedení některých symbolů (matematických) atd. Hlavním cílem tohoto kroku je pokusit se takové chyby opravit, odfiltrovat neznámé znaky (vyčistit text) a upravit text do podoby, která je vhodnější pro budoucí zpracování. Nejdůležitější kroky předzpracování textu které provádím jsou:

1. odstranění zbytečných znaků – filtrace všech znaků mimo znaky anglické abecedy, bílé znaky, vybraná interpunkční znaménka a čísla

2. spojení slova rozděleného znakem rozdělovníku mezi více řádků – některá slova vyskytující se na konci řádku jsou rozděleny znakem rozdělovníku na začátek dalšího řádku, je potřeba taková slova spojit

3. záměna všech bílých znaků (i vícenásobných) za jednu mezeru – všechna slova po tomto kroku budou oddělena pouze jednou mezerou a text se tímto pročistí
4. oddělení slov od interpunkčních znamének na konci věty a souvětí – interpunkční znaménko na konci věty nebo souvětí je vždy spojeno s posledním vyskytujícím se slovem, je potřeba znaménko oddělit mezerou
5. odstranění číslovek – odfiltrování samostatně se vyskytujících čísel
6. převod na malé znaky – posledním krokem je převod textu na malé znaky

Konečným výsledkem by měl být text, obsahující slova oddělena mezerou. V případě konce věty, nebo souvětí pak oddělena mezerou, daným interpunkčním znaménkem a další mezerou. Všechny znaky jsou malé a text neobsahuje neznámé a nepoužívané znaky a samostané číslovky.

7.2.3 Výběr kandidátních termínu a slovních spojení

V tomto kroku se předzpracovaný text rozdělí na jednotlivé kandidátní výrazy. Výrazů v každé publikaci však může být opravdu mnoho. Příliš mnoho výrazů má za následek určité zpomalení systému v dalších krocích zpracování, protože se nad každým výrazem provádějí různé výpočty a porovnávání. Proto je dobré některá nevhodná slova určitým způsobem odfiltrovat, což povede k následnému zrychlení. Rozhodl jsem se pro 3 druhy filtrace:

- odfiltrování krátkých slov
- použití stoplistu
- použití morfologického analyzátoru

První nejjednodušší variantou je odfiltrovat všechna krátká slova. Zjistil jsem, že relativně často se v textech vyskytují slova o délce 1-3 znaky. Když se však podíváme na výsledky experimentu 6.1, můžeme vidět, že mezi klíčovými výrazy se takto dlouhá slova vyskytuji v 7% případů. Vzhledem k možnému vysokému nárůstu vygenerovaných kandidátních výrazů je však toto číslo relativně zanedbatelné.

Dalším řešením filtrace kandidátů je využití stoplistu. Stoplist obsahuje nejčastěji používaná slova v daném jazyce (předložky, spojky, některá podstatná a přídavná jména atd.). Je jasné, že tato slova s největší pravděpodobností nebudou zařazena mezi klíčová. Odstraněním takových slov z textu docílíme další výrazné redukce možných kandidátních výrazů. Použití stoplistu je v mém systému volitelné.

Posledním přístupem k filtraci je použití morfologického analyzátoru. V mém systému používám morfologický analyzátor TreeTagger popsaný v kapitole 5.2. Morfologický analyzátor dokáže určit slovní druh a základní tvar každého slova ve vstupním textu. Je proto možné odfiltrovat nepoužívané slovní druhy. Jak ukázal experiment 6.5, jasně nejpoužívanějším slovním druhem je podstatné jméno. Jako další následují přídavná jména a slovesa v gerundiu. Posledním druhem co stojí za zmínku jsou vlastní jména a názvy.

Jiným možným přístupem k filtraci s využitím morfologického analyzátoru je vyhledávání vzorů slovních druhů v textu. Právě takový přístup používám v mém systému. Jednotlivá slova v textu převedu na slovní druhy a poté v nich vyhledávám nejpoužívanější vzory klíčových výrazů. Ty jsem se pokusil najít v experimentu 6.6. Tato varianta vyhledávání kandidátních termínu je v systému volitelná.

Další výhodou použití morfologického analyzátoru je možnost převodu slov do základního tvaru (lemma). Některé výrazy se v textu mohou vyskytovat několikrát, pokaždé však v jiném tvaru. Lemmatizace je důležitá ze dvou důvodů:

- přehlednější nabídka kandidátních výrazů v základním tvaru
- sjednocení výrazů v různých podobných tvarech do jednoho

Pokud není v systému k výběru kandidátních termínů použit morfologický analyzátor, následuje po filtraci samotné spojování slov do výrazů. Vyzkoušel jsem spojování relevantních slov na základě výpočtu směrodatné odchylky (kapitola 2.1). Ve výsledku se však generovalo mnoho zbytečných a příliš nesouvisejících kandidátních výrazů. Proto jsem se rozhodl jít jednoduchou cestou a používám klasické spojování sousedních slov do výrazu. Je však potřeba omezit maximální délku výrazu (počet slov). Experiment 6.2 ukázal, že více než 98% všech klíčových výrazů má délku 1-3 slova. Některé výrazy sice mohou být delší, není jich však mnoho. Proto jsem se rozhodl omezit výběr pouze na unigramy, bigramy a trigramy.

7.2.4 Porovnání s backgroundovým korpusem

V tomto kroku se nalezené kandidátní výrazy porovnají s výrazy v backgroundových datech. Porovnáním je myšleno nalezení výrazu v backgroundu, zjištění jeho četnosti a počet článků obsahujících výraz. Background se skládá z velkého množství výrazů vyextrahovaných z článků podobného typu. Já si vytvořil 2 druhy backgroundu:

- backgroundová data z konference
- backgroundová data z nejpodobnějších článků

Backgroundová data z konference se skládají z výrazů vyextrahovaných z článků jedné konference. Předpokládá se, že články publikované na jedné konferenci by měly patřit do stejné významové domény. Vytvořil jsem tedy backgroundová data pro všechny konference z mých testovacích dat zvlášť.

Druhým typem je background vytvořený z množiny nejpodobnějších článků z celé kolekce testovacích dat. Každý článek má své vlastní backgroundová data. Bylo tedy potřeba nalézt sémanticky nejpodobnější dokumenty ke každému článku. K tomuto účelu jsem použil nástroj *Related docs* popsaný v 5.3. Tento nástroj slouží k zjišťování sémantické blízkosti mezi dokumenty. Proces výpočtu je časově velmi náročný, proto je potřeba mít data předzpracovaná.

Tvorba backgroundu probíhá tak, že se vyberou všechny výrazy ze všech článků patřících do množiny ze kterých se background vytváří. Poté se vypočítá četnost každého nalezeného výrazu v rámci celé množiny článků. Při výpočtu se také sečtou všechny dokumenty ve kterých se výraz objevuje. Nalezené výrazy, jejich četnost a počet dokumentů se poté uloží do souboru. Všechna potřebná backgroundová data jsem si také předzpracoval.

7.2.5 Ohodnocení termínů

Posledním krokem je samotný výběr nejdůležitějších klíčových slov. To probíhá na základě ohodnocení výrazů pomoci některé statistické metody. Rozhodl jsem se implementovat následující metody:

- *Term Frequency* (kap. 3.3)

- *Term Frequency - Inverse Document Frequency* (kap. 3.4)
- *Residual IDF* (kap. 3.5)

Podle ohodnocení vybranou metodou se kandidátní výrazy seřadí. Nejlépe ohodnocené výrazy jsou označeny jako klíčové.

7.3 Implementace systému

Jako implementační jazyk jsem zvolil skriptovací jazyk Python (popsaný v kapitole 5.1). Systém jsem se rozhodl rozdělit do 3 modulů. Každá část je nezávislá a vykonává určité části zpracování popsané v předchozí kapitole. Výhodou je, že se jednotlivé kroky zpracovávají postupně, každý skript generuje vlastní výstup. Skripty se mohou použít v kombinaci s jinými, mohou se vzájemně kombinovat. Při dodržení struktury vstupních dat je možno použít jako vstup některé části data generovaná zcela nezávislým systémem. Na obrázku 7.2 můžete vidět návaznost jednotlivých modulů.

Při implementaci jsem se také zaměřil na hromadné zpracování více článků současně. Hlavním vstupem všech skriptů je XML soubor s definovanou strukturou, obsahující většinou cesty k samotným článkům. To při hromadném zpracování uživateli značně ulehčuje práci.

Obrázek 7.2: Struktura implementovaných modulů

Prvním skriptem je *parser.py*. Ten po načtení publikací v textové podobě články postupně předzpracuje a vyhledá v nich všechny kandidátní výrazy. Předzpracování probíhá pomoci regulárních výrazů. Pro výběr kandidátních výrazů si uživatel může zvolit, zda chce použít stoplist. Stoplist může být libovolný soubor se seznamem slov oddělenými koncem řádku. Další volbou uživatele je také použití morfologického analyzátoru. Hlavním výstupem tohoto skriptu je seznam všech kandidátních výrazů každého zpracovaného článku s četností výskytu výrazu v článku.

Druhým skriptem systému je *background.py*. Tento skript má za úkol porovnat kandidátní výrazy s backgroundovými daty. Skript se každý načtený kandidátní výraz pokusí vyhledat v backgroundu. Pokud je nalezen, zjistí se jeho četnost a počet dokumentů v kterých se vyskytuje. Poté se tyto údaje zapíší do výstupního souboru.

Backgroundových dat může být několik typů. Uživatel si může vybrat, zda chce pro každý článek používat jiný background vytvořený z nejpodobnějších článků, nebo zda chce používat jiný background pro každou konferenci. Poslední možností je načtení jedno společného backgroundu pro všechny současně zpracovávané články.

Poslední částí systému je skript *methods.py*. Jeho hlavní činností je výpočet ohodnocení všech kandidátních výrazů podle uživatelem zvolené metody. Výrazy jsou poté podle tohoto ohodnocení seřazeny a vypsány do výstupního souboru. Výrazy z nejvyšším ohodnocením jsou označeny jako klíčové.

Tvorba stoplistu

Stoplist je důležitý k odfiltrování nežádoucích a nedůležitých slov z textu. Odfiltrováním těchto slov zabráníme generování zbytečných kandidátních výrazů, které zcela jistě nebudou patřit ke klíčovým. Tím dosáhneme určitého zrychlení našeho systému, jelikož nebude potřeba provádět výpočty nad některými výrazy. Většinou se jedná o obecná slova často používaná v běžném jazyce.

K tvorbě stoplistu jsem si vytvořil seznam nejčastěji používaných slov v mých článcích a jejich četnosti. Následně jsem si vytvořil stejný seznam pro všechny klíčové výrazy. Poté jsem ručně procházel nejčastější slova z článků a zjišťoval jejich četnost v klíčových výrazech. Pokud byla četnost nižší než určitá mez, přidal jsem slovo do stoplistu. Výsledkem je stoplist, který má celkem 114 slov.

Podobně jsem postupoval při tvorbě stoplistu pro lemmatizované články. Články a klíčové výrazy jsem musel lemmatizovat. Tento stoplist má pouze 102 výrazů. Některé slova se totiž sloučili do jednoho (například *be – was – were*).

7.4 Ovládání

V této části se pokusím popsat ovládání implementovaného systému. Popíšu postupně způsob ovládání a vlastnosti jednotlivých skriptů, jejich vstupy a výstupy a také strukturu podporovaných datových souborů.

7.4.1 parser.py

Tento skript načte články uvedené ve vstupním XML souboru, a vyhledá v nich všechny termíny a výrazy. Výsledek zapíše do souborů (pro každý článek zvlášť) a vytvoří výstupní XML soubor.

Uživatel má na výběr, zda bude chtít při filtraci použít stoplist, či zda bude aktivován morfologický analyzátor. Má také možnost zpracovat pouze omezený počet článků ze vstupního souboru. Uživatel si může zvolit výběr článků náhodným výběrem. Poslední možností je volba názvu výstupního XML souboru. Při použití morfologického analyzátoru budou nalezené kandidátní termíny v základním tvaru.

Ke správné funkci skriptu je potřeba nastavit proměnné prostředí. Nastavení naleznete v příloze B.

Popis parametrů

```
parser.py -f INPUT [-r COUNT] [-s STOPLIST] [-t] [-l ID NUM] [-o OUTPUT]

 -f INPUT     Vstupní XML soubor.
 -r COUNT     Budou se zpracovávat pouze náhodné články. Počet udává COUNT.
 -s STOPLIST  Načtení stoplistu. STOPLIST udává cestu. Pokud není zadáno,
              stoplist nebude použit.
 -t           Bude se využívat morfologický analyzátor.
 -l ID COUNT  Limit zpracování článků. Zpracuje se pouze COUNT článků od ID.
              První článek je brán jako ID 0.
 -o OUTPUT    Výstupní XML soubor. Pokud není zadáno, výstup se uloží do
              "parser.xml".
 -h           Nápověda.
```

Struktura vstupního XML souborů

```
<clanky>
    <clanek>
        <konference>konference</konference>
        <nazev>nazev.txt</nazev>
    </clanek>
    <clanek>
        <konference>konference2</konference>
        <nazev>nazev2.txt</nazev>
    </clanek>
</clanky>
```

Předpokládá se, že článek je uložen v souboru `konference/txt/nazev.txt`.

Struktura výstupního XML souborů

```
<clanky>
    <clanek>
        <konference>konference</konference>
        <nazev>nazev.txt</nazev>
        <cesta>konference/terms/nazev.txt.terms</cesta>
    </clanek>
</clanky>
```

Do tagu `<cesta>` je vložena cesta k vytvořenému souboru se všemi nalezenými výrazy.

Struktura výstupního souboru s výrazy

Výstupní soubory s kandidátními výrazy se vždy ukládají do adresáře `terms` v adresáři konference. K názvu článku se přiřadí přípona `.terms`. Cesta tedy může vypadat například takto: `konference/terms/nazev.txt.terms`

Struktura takového souboru je velmi jednoduchá. Každý řádek obsahuje jeden výraz. Výraz je oddělen tabulátorem nasledovaným počtem výskytů tohoto výrazu v článku. Příkladem může být tento soubor:

```
vyraz1    10
vyraz2 vyraz2    2
vyraz3    5
vyraz4 vyraz4 vyraz4    3
```

7.4.2 background.py

Tento skript porovná nalezené výrazy v předchozího kroku s backgroundovými daty. Skript se každý kandidátní výraz pokusí vyhledat v backgroundu. Pokud je nalezen, zjistí se jeho četnost a počet dokumentů v kterých se vyskytuje. Poté se tyto údaje zapíší do výstupního souboru.

Uživatel musí zadat cestu ke vstupnímu XML souboru, odkud se načtou cesty ke kandidátním výrazům. Další povinnou volbou je výběr typu backgroundových dat. Uživatel má na výběr backgroundová data konference, backgroundová data nejpodobnějších článků,

nebo společný background pro všechny články. První dvě možnosti mají pevná pravidla názvu a cesty k backgroundům. U poslední možnosti musí uživatel zadat cestu k backgroundu ručně. Poslední nepovinnou možností je volba názvu výstupního XML souboru.

Popis parametrů

```
background.py -f INPUT [ -k | -s | -b BACKGROUND ] [-o OUTPUT]

 -f INPUT       Vstupní XML soubor.
 -b BACKGROUND  Bude použit společný background. BACKGROUND udává cestu.
 -s             Bude použit background pro každý článek zvlášť.
 -k             Bude použit background pro každou konferenci zvlášť.
 -o OUTPUT      Výstupní XML soubor. Pokud není zadáno, výstup se uloží
                do "background.xml".
 -h             Nápověda.
```

Pokud bude vybrán parametr `-s`, musí být background každého článku uložen v souboru `konference/background/nazev.txt.back`. Jestliže bude vybrán parametr `-k`, předpokládá se, že bude background uložen v `konference/nazev_konference.back`.

Struktura vstupního XML souborů

Struktura vstupního XML souboru je stejná jako výstup předchozího skriptu (viz 7.4.1).

Struktura výstupního XML souborů

```
<clanky>
    <clanek>
        <nazev>nazev.txt</nazev>
        <konference>konference</konference>
        <cesta>konference/terms/nazev.txt.back.terms</cesta>
        <background>pocet_clanku</background>
    </clanek>
</clanky>
```

V tagu `<cesta>` je uložena cesta k vytvořenému souboru obsahujícím výrazy s jednotlivými počty výskytů. Tag `<background>` obsahuje počet článků v použitém backgroundu.

Struktura výstupního souboru s výrazy

Výstupní soubory se stejně jako v předchozím skriptu ukládání do adresáře `terms` v adresáři konference. K názvu článku se přiřadí přípona `.back.terms`. Cesta tedy může vypadat například takto: `konference/terms/nazev.txt.back.terms`

Struktura souboru je také velmi podobná předchozímu. Každý řádek obsahuje jeden výraz, který je oddělen tabulátorem nasledovaným počtem výskytů tohoto výrazu v článku. Navíc však přibyly dva sloupce opět odděleny tabulátorem. Prvním je celkový počet výskytů tohoto výrazu v daném backgroundu. Druhým je pak počet dokumentů, ve kterých se výraz nachází. Příkladem může být tento soubor:

```
vyraz1    10    107    38
vyraz2 vyraz2    2    10    5
vyraz3    5    123     52
vyraz4 vyraz4 vyraz4    3    0    0
```

Struktura souboru s backgroundovými daty

Soubor s backgroundovými daty je textový soubor obsahující výrazy seřazené podle četnosti výskytů. Na prvním řádku tohoto souboru najdeme číslo značící počet článků, ze kterých byl background sestaven.

Další řádky pak již obsahují jednotlivé výrazy. Na každém řádku je jeden výraz. Ten následuje sloupec s celkovým počtem výskytů tohoto výrazu v daném backgroundu. Posledním sloupcem je pak počet dokumentů obsahující daný výraz. Sloupce jsou odděleny tabulátorem. Příklad takového souboru:

```
95
vyraz1    107    38
vyraz2 vyraz2    10    5
vyraz3    123     52
```

7.4.3 methods.py

Poslední skriptem je `methods.py`. Jeho hlavní činností je výpočet ohodnocení všech kandidátních výrazů podle uživatelem zvolené metody. Výrazy jsou poté podle tohoto ohodnocení seřazeny a vypsány do výstupního souboru (pro každý článek zvlášť). Výrazy s nejvyšším ohodnocením jsou označeny jako klíčové.

Uživatel musí zadat cestu ke vstupnímu XML souboru. Ten obsahuje cesty k souborům s termíny. Další povinnou volbou je výběr metody ohodnocení. Má možnost zvolit *Term Frequency* (3.1), *Term Frequency - Inverse Document Frequency* (3.4), nebo *Residual IDF* (3.5). Uživatel také může zadat limit počtu klíčových slov zapsaných do výstupního XML souboru. Pokud limit zadán nebude, do XML se žádný výraz nevypíše. Poslední nepovinnou možností je volba názvu výstupního XML souboru.

Popis parametrů

```
methods.py -f INPUT -m METHOD [ -l LIMIT ] [ -o OUTPUT ]

 -f INPUT   Vstupní XML soubor.
 -m METHOD  Výběr metody ohodnocení. METHOD = [ tf | tfidf | ridf ]
 -l LIMIT   Limit počtu klíčových slov zapsaných ve výstupním XML.
 -o OUTPUT  Výstupní XML soubor. Pokud není zadáno, výstup se uloží
            do METHOD.xml.
 -h         Nápověda.
```

Struktura vstupního XML souborů

Struktura vstupního XML souboru je stejná jako výstup předchozího skriptu (viz 7.4.2).

Struktura výstupního XML souborů

```
<clanky>
    <clanek>
        <nazev>nazev.txt</nazev>
        <konference>konference</konference>
        <cesta>konference/terms/nazev.txt.tfidf</cesta>
        <terminu>pocet_terminu_v_clanku</terminu>
        <tagy>
            <termin>termin1</termin>
            <termin>termin2</termin>
            <termin>termin3</termin>
        </tagy>
    </clanek>
</clanky>
```

Pokud není zadán název výstupního XML souboru, je pojmenován podle uživatelem vybrané metody (pro metodu *tf-idf* se vytvoří `tfidf.xml`) V tagu `<cesta>` je uložena cesta k nově vytvořenému souboru obsahujícím výrazy s ohodnocením. Výrazy jsou seřazeny podle tohoto ohodnocení. Tag `<terminu>` obsahuje počet všech analyzovaných kandidátních termínů v článku.

Struktura výstupního souboru s ohodnocenými výrazy

Výstupní soubory se stejně jako v předchozích skriptech ukládání do adresáře `terms` v adresáři konference. K názvu článku se přiřadí přípona s názvem ohodnocovací metody (pro *tf-idf* je to `.tfidf`). Celková cesta tedy může vypadat například takto:
`konference/terms/nazev.txt.tfidf`.

Každý řádek toho souboru obsahuje jeden výraz nasledovaný velikosti ohodnocení odděleným tabulátorem. Výrazy jsou seřazeny podle nejvyššího ohodnocení. Příkladem může být tento soubor:

```
vyraz4 vyraz4 vyraz4    0.0547518930368
vyraz3    0.0188381748603
vyraz1    0.0175385138918
vyraz2 vyraz2    0.0168603409893
```

Kapitola 8

Výsledky

Cílem této kapitoly bude pokusit se ohodnotit kvalitu výsledného systému. Na kvalitu se dá pohlížet z několika hledisek. Provedl jsem tedy několik testů nad testovacími články. V prvním testu zjišťuji, zda vytvořený systém klíčový výraz vůbec nalezl, popřípadě z jakého důvod nebyl výraz nalezen.

V dalším testu se pokusím zjistit, jakou souvislost má nastavení parseru na počet vybraných kandidátních výrazů a na počet správně vybraných klíčových slov. Dále se pokusím zjistit úspěšnost ohodnocení klíčových výrazů. K testování jsem použil 2 typy backgroundu. V poslední části analyzuji výstupy mého systému na zvoleném článku.

8.1 Identifikované klíčové výrazy

V prvním testu se pokouším zjistit, zda můj systém vůbec identifikoval klíčové výrazy v článku. Pokusím se popsat problémy, ke kterým při nenalezení výrazu mohlo dojít a tyto problémy se u takových výrazů pokusit identifikovat. Nakonec pak vytvořit statistiku těchto problémů.

Metodika testování byla taková, že jsem hledal klíčové výrazy v článku s různými kombinacemi nastavení skriptu `parser.py` – použití morfologického analyzátoru, použití stoplistu a nastavení úvodní filtrace. Poté jsem jednotlivé výstupy porovnal. Narazil jsem na několik variant problémů při identifikaci výrazů:

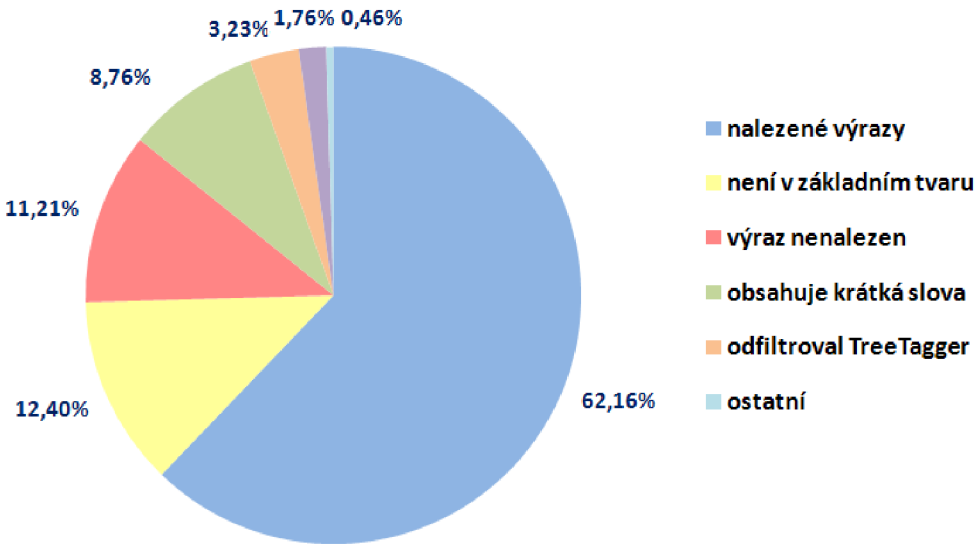
- Výraz byl nalezen, všechno v pořádku.
- Hledaný výraz se v textu nevyskytuje v základním tvaru. Při použití morfologického analyzátoru a převodu do základního tvaru je identifikován bez problému.
- TreeTagger výraz odfiltroval. Znamená to, že výraz je složen z netypických spojení slovních druhů, které se pro klíčová slova příliš nepoužívají.
- Některá část výrazu byla odfiltrovaná stoplistem. To znamená, že výraz obsahuje některé slovo vyskytující se v použitém stoplistu.
- Výraz se v článku vyskytuje, je však v jiném tvaru. V tomto případě klíčové slovo není v základním tvaru a v textu se vyskytuje ještě v jiném tvaru.
- Výraz se skládá z více než 3 slov, proto byl odfiltrován.
- Některé slovo ve výrazu je kratší než 4 znaky, proto bylo odfiltrováno.

- Výraz obsahuje některé znaky, které byly odfiltrovány již v prvotním zpracování článku. Jedná se o znaky, které nepatří do anglické abecedy, nejsou to vybraná interpunkční znaménka ani čísla.
- Nedefinovaný problém. Výraz se v článku nachází, nebyl však hodnocen. V tomto případě jsem nedokázal automatizovaným způsobem určit proč tomu tak je.
- Výraz se v článku vůbec nevyskytuje.

V tabulce 8.1 je shrnuta statistika těchto problémů identifikace. V prvním sloupci je popsán typ problému, druhý sloupec obsahuje počet výrazů, v posledním sloupci najdete vyjádření v procentech. Tabulku 8.1 jsem pro přehlednost znázornil také do grafu 8.1.

typ problému	výrazů	v procentech
výraz byl nalezen	24 518	62,16%
výraz není v základním tvaru	4 892	12,40%
výraz odfiltrován TreeTaggerem	1 273	3,28%
výraz odfiltrován stoplistem	39	0,1%
výraz se vyskytuje v jiném tvaru	56	0,14%
výraz se skládá z více než 3 slov	696	1,76%
některé slovo kratší než 4 znaky	3 456	8,76%
odfiltrováno v prvotním zpracování	10	0,02%
nedefinovaný problém	78	0,2%
výraz se v článku nevyskytuje	4 422	11,21%
CELKEM VÝRAZŮ	39 440	100%

Tabulka 8.1: Tabulka problémů identifikace klíčových výrazů.



Obrázek 8.1: Graf identifikovaných klíčových výrazů.

Z tabulky a z grafu můžeme vyčíst, že 62% všech vzorových klíčových výrazů se nachází v textu v základním tvaru a byly systémem bez problému nalezeny. Okolo 12% výrazů se v textu nachází v jiném tvaru, klíčový výraz však v základním tvaru je. Takové výrazy je možné s použitím morfologického analyzátoru a převodu textu do základního tvaru bez problému identifikovat. Téměř 9% výrazů obsahuje slova kratší délky 4, proto nebyly identifikovány. Je to docela vysoká hodnota, ale vzhledem k možnému velkému nárůstu kandidátních klíčových výrazů je tento filtr zapotřebí. Zhruba 3% výrazů odfiltroval TreeTagger. Jejich skladba slovních druhů totiž není příliš typická pro klíčové výrazy.

Všechny případy filtrace výrazů popsaných výše se dají ovlivnit různým nastavením filtrů, použitím morfologického analyzátoru, či úpravou stoplistu. Naopak 11,21% výrazů se v článcích nenachází vůbec. Není je proto možné bez dalších podpůrných dat identifikovat žádným způsobem. V experimentu 6.4 jsem se také pokusil vyhledávat klíčová slova v testovacích článcích, nebyl však použit morfologický analyzátor. Počet nenalezených výrazů se oproti tomuto experimentu snížil o více než polovinu.

V tomto testu jsem se také pokusil prozkoumat, zda jsou tyto výsledky závislé na zdroji článků (na konferenci). Zjistil jsem, že výsledky ve značné míře zkreslují články Oxford Journals, PNAS a LREC, kde se počet v článku nevyskytujících se výrazů pohybuje okolo 20% (PNAS dokonce 25%). U zhruba poloviny konferencí se toto číslo pohybuje hluboko pod 5%. Počet nalezených výrazů se s vyjímkou těchto 3 zdrojů pohybují okolo 70%.

8.2 Výběr kandidátních výrazů

V tomto testu se pokusím zjistit, jakou souvislost má nastavení parseru na počet vybraných kandidátních výrazů. Také se pokusím zjistit, zda se redukcí počtu kandidátních výrazů odfiltrují také některá vhodná klíčová slova. Tento test je důležitý z hlediska toho, zda při redukci počtu kandidátních výrazů nedochází také k odstranění důležitých výrazů, které by mohli být klíčovými.

V první části testu bylo potřeba vytvořit statistiku počtu vybraných kandidátních výrazů s různým nastavením parseru. Nastavení parseru ovlivní výběr kandidátních výrazů. Použil jsem 3 typy nastavení. Jedná se o kombinace použití stoplistu a morfologického analyzátoru.

Nastavení parseru:

- základní filtr – při výběru kandidátních výrazů byl použit pouze základní filtr, odstraňující nevhodné znaky, krátká slova a výrazy delší než 3 slova
- se stoplistem – při výběru kandidátních výrazů byl použit k filtraci základní stoplist odstraňující nevhodná slova
- s TreeTaggerem – při výběru kandidátních výrazů byl použit k filtraci stoplist a morfologický analyzátor TreeTagger

Výsledky jsou zapsány v tabulce 8.2. Jelikož se výsledky pro různé datové zdroje mohou lišit, vytvořil jsem statistiku pro každou konferenci samostatně. V prvním sloupci je název konference. Další sloupce obsahují průměrný počet vybraných kandidátních výrazů na článek. V každém sloupci je výsledek pro jiné nastavení parseru – s použitím TreeTaggeru, se stoplistem, pouze se základním filtrem.

KONFERENCE	s TreeTaggerem	se stoplistem	základní filtr
ASRU	1 126	1 597	2 186
ICASSP	775	1 069	1 422
ICSLP	880	1 231	1 694
EMBS	484	641	850
Coling	1 020	1 458	1 988
LREC	1 218	1 680	2 210
GWN	984	1 364	1 862
PNAS	1 421	1 984	2 693
Oxford Journals	1 681	2 368	3 221
CELKEM	1 133	1 581	2 141

Tabulka 8.2: Tabulka počtu kandidátních výrazů.

Z výsledků můžeme vyčíst, že nejvíce kandidátních výrazů vybráno s použitím základního filtru, což bylo jasné. Průměrně je to 2 141 výrazů na článek. Výsledky se u některých konferencí hodně liší. Důvodem je různá délka článků.

Při použití stoplistu se počet kandidátních výrazů docela výrazně zredukuje. Průměrná hodnota je 1 581 výrazů na článek. Znamená to snížení počtu o zhruba 26%. Ve všech konferencích je tato redukce přibližně stejná.

K další redukci kandidátních výrazů došlo použitím morfologického analyzátoru. Průměrný počet se snížil na 1 133, což je o zhruba 30% méně než při použití stoplistu a dokonce o 47% méně než při použití základního filtru.

V druhé části tohoto testu jsem se pokusil zjistit, zda se touto redukcí neodstraní z výběru kandidátních výrazů také vhodná klíčová slova. Testování probíhalo tak, že jsem vybral kandidátní výrazy s určitým nastavením parseru a zkoumal jsem, zda se mezi těmito výrazy nacházejí klíčová slova přiřazené k danému článku. Test jsem opět provedl pro každou konferenci zvlášť.

Výsledky toho testu najdete v tabulce 8.3. První sloupec obsahuje název konference (datového zdroje). Ve druhém sloupci najdete celkový počet klíčových slov. Další tři sloupce zobrazují počet nalezených klíčových slov mezi kandidátními výrazy pro určité nastavení parseru.

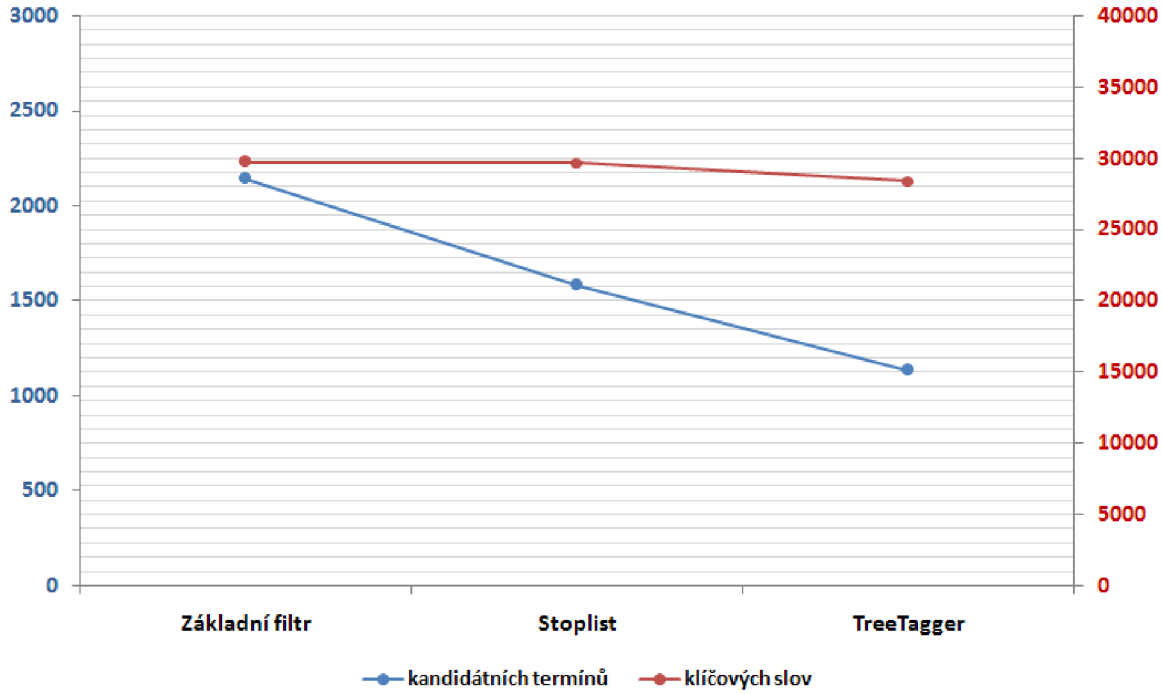
KONFERENCE	výrazů	s TreeTaggerem	se stoplistem	základní filtr
ASRU	391	331 — 83%	336 — 85%	334 — 85%
ICASSP	8 926	7 494 — 83%	7 761 — 87%	7 775 — 87%
ICSLP	7 061	5 988 — 84%	6 171 — 87%	6 176 — 87%
EMBS	4 211	3 426 — 81%	3 506 — 83%	3 511 — 83%
Coling	490	390 — 78%	392 — 79%	392 — 80%
LREC	3 812	2 573 — 67%	2 563 — 67%	2 566 — 67%
GWN	116	95 — 81%	98 — 82%	96 — 82%
PNAS	5 874	3 360 — 56%	3 562 — 60%	3 567 — 60%
Oxford Journals	8 559	4 983 — 57%	5 273 — 61%	5 284 — 61%
CELKEM	39 440	28 389 — 72%	29 662 — 75%	29 701 — 75%

Tabulka 8.3: Tabulka nalezených výrazů.

Z tabulky můžeme vyčíst, že při použití základního filtru bylo mezi kandidátními výrazy průměrně nalezeno 75% klíčových slov. Pokud do parseru přidáme filtraci pomoci stoplistu, tato hodnota zůstane téměř nezměněna. Jak jsem však ukázal v první části testu (tabulka 8.2), počet kandidátních výrazů se při použití stoplistu zredukuje o 26%, což není zanedbatelná hodnota. Využití stoplistu je tedy velmi výhodné.

Pokud při výběru použijeme morfologický analyzátor se stoplistem dohromady, počet nalezených klíčových slov se sníží o zhruba 3% oproti základnímu filtru na 72%. Počet vybraných kandidátních výrazů se však sníží o 47%, což je skoro polovina. Rozdíl je tedy velmi vysoký a je výhodné použít také morfologický analyzátor. Navíc převede všechny výrazy do základního tvaru. Sloučí se tedy všechny příbuzné termíny. Toho se poté využije k lepšímu ohodnocení výrazu.

Výsledek redukce počtu kandidátních výrazů a počtu klíčových slov jsem se pokusil znázornit v grafu 8.2.



Obrázek 8.2: Graf závislosti počtu kandidátních výrazů a klíčových slov.

8.3 Úspěšnost ohodnocení výrazů

V tomto testu jsem se pokusil porovnat úspěšnost různých metod ohodnocení nalezených kandidátních výrazů. Test jsem opět provedl s různým nastavením systému. Vyzkoušel jsem různé varianty parseru a také 2 druhy backgroundu.

Použil jsem stejné nastavení parseru jako u předchozího testu:

- základní filtr

- se stoplistem
- s TreeTaggerem

Typy backgroundu:

- background vytvořený z článků konference
- background vytvořený z množiny nejpodobnějších článků

Testování probíhalo tak, že jsem vypočítal průměrnou pozici klíčového slova v seřazené posloupnosti všech kandidátních výrazů podle ohodnocení určitou metodou. Jelikož má každý článek různý počet kandidátních výrazů, bylo potřeba výsledek normalizovat. Podělil jsem tedy spočtenou hodnotu počtem kandidátních výrazů podle vzorce:

$$rank = \frac{prumerna\ pozice\ vyrazu}{pocet\ kandidatnich\ vyrazu} \times 100. \tag{8.1}$$

Výsledkem je průměrná pozice výrazu v procentech. Pokud je výsledkem např. 20% a článek má 100 kandidátních výrazů, znamená to, že průměrný klíčový výraz leží na 20. pozici. Z toho vyplývá že čím nižší je tato hodnota, tím byla metoda ohodnocení úspěšnější.

V testu jsem použil všechny 3 implementované metody ohodnocení (*tf*, *tf-idf* a *ridf*) a výpočet jsem provedl pro každou konferenci zvlášť. Výsledky můžete vidět v tabulce 8.4 pro background z nejpodobnějších článků a v tabulce 8.5 pro background vytvořený z článků konference. První sloupec tabulky je název konference. V dalších sloupcích najdeme výsledky pro různé nastavení parseru. Každý sloupec obsahuje 3 hodnoty. Hodnota udává výsledek pro jednu metodu hodnocení.

KONFERENCE	s TreeTaggerem *tfidf* — *ridf* — *tf*	se stoplistem *tfidf* — *ridf* — *tf*	základní filtr *tfidf* — *ridf* — *tf*
ASRU	17% — 17% — 18%	19% — 18% — 19%	29% — 29% — 25%
ICASSP	17% — 17% — 20%	19% — 19% — 23%	25% — 26% — 29%
ICSLP	16% — 17% — 17%	18% — 18% — 21%	25% — 26% — 25%
EMBS	16% — 17% — 19%	18% — 19% — 22%	25% — 27% — 27%
Coling	16% — 17% — 19%	20% — 19% — 19%	25% — 25% — 22%
LREC	19% — 20% — 20%	21% — 21% — 24%	31% — 31% — 27%
GWN	17% — 16% — 18%	18% — 15% — 20%	27% — 25% — 25%
PNAS	22% — 23% — 21%	27% — 28% — 26%	39% — 42% — 31%
Oxford Journals	23% — 24% — 20%	26% — 27% — 24%	38% — 42% — 28%
CELKEM	19% — 20% — 20%	21% — 21% — 23%	29% — 31% — 28%

Tabulka 8.4: Background složený z množiny podobných článků.

Při použití základního filtru byly podle předpokladu v obou případech výsledky nejhorší. Systém se stoplistem dosahoval výrazně lepších výsledků, než systém se základním filtrem (asi o 8%). Nejlépe podle předpokladu dopadl systém s morfologickým analyzátorem. Rozdíl oproti systému se stoplistem však již příliš vysoký nebyl (asi 2%).

Nejlepší metodou podle testu je *tf-idf*, která nejlépe vyhodnocovala výrazy v systému se stoplistem i v systému s morfologickým analyzátorem v obou tabulkách. V těchto případech byla podle předpokladu nejhorší metoda *tf*. Zajímavé však je, že metoda *tf* byla nejlepší pro

KONFERENCE	s TreeTaggerem *tfidf* — *ridf* — *tf*	se stoplistem *tfidf* — *ridf* — *tf*	základní filtr *tfidf* — *ridf* — *tf*
ASRU	17% — 18% — 18%	18% — 19% — 19%	25% — 25% — 25%
ICASSP	18% — 18% — 20%	20% — 20% — 23%	28% — 22% — 29%
ICSLP	17% — 17% — 18%	17% — 18% — 21%	24% — 20% — 25%
EMBS	16% — 16% — 19%	18% — 18% — 22%	25% — 20% — 27%
Coling	16% — 17% — 19%	19% — 18% — 19%	24% — 24% — 22%
LREC	20% — 20% — 20%	21% — 22% — 23%	31% — 29% — 27%
GWN	22% — 17% — 18%	19% — 16% — 20%	29% — 32% — 25%
PNAS	20% — 20% — 21%	25% — 25% — 26%	33% — 31% — 31%
Oxford Journals	20% — 21% — 20%	22% — 23% — 24%	33% — 30% — 28%
CELKEM	18% — 19% — 20%	20% — 21% — 23%	28% — 25% — 28%

Tabulka 8.5: Background složený z článků konference.

systém se základním filtrem. Další zajímavostí je, že při použití metody *ridf* a backgroundu složených z konferencí byl výsledek o 6% lepší než při použití stejné metody s backgroundem složeného z nejpodobnějších článků.

Nejhůře hodnoceny byly ve všech případech články PNAS a Oxford Journals, které vzhledem k počtu článků také mohly negativně ovlivnit celkový průměrný výsledek. Naopak nejkvalitněji byly ohodnoceny články konference EMBS.

Obě tabulky ukazují, že se výsledky při použití těchto druhů testovaných backgroundů od sebe nijak výrazně neliší. V nejlepších případech dosahovala průměrná pozice klíčového slova 16%. Tato hodnota není nikterak závratná, musíme však počítat s určitým zkreslením výsledku daným některými klíčovými výrazy, které byly ohodnoceny velmi špatně. Jedná se hlavně o výrazy, které byly označeny v testovacích datech jako klíčové, přitom jsou však velmi obecné a používají se v textu mnoha článků. Takové výrazy přímo nevystihují daný článek. Typickým příkladem může být výraz *speech recognition*, který se v mnoha článcích vyskytuje jako klíčový, přitom o rozpoznávání řeči obecně pojednává celá konference.

Řekněme, že máme 100 kandidátních výrazů a klíčová slova seřazené podle ohodnocení se vyskytují na pozici 1, 3 a 92. Poslední výraz je právě ten obecný. Průměrné ohodnocení je v tomto případě 32%, což se na první pohled může zdát jako vysoká hodnota, výrazy však byly ohodnoceny relativně dobře.

8.4 Analýza systémem vybraných klíčových výrazů

Doposud jsem přiřazené klíčové termíny k článkům bral jako jedinou možnost správně vybraných klíčových výrazů. Je jistě pravda, že tyto termíny jsou vybrány správně. Vybíral je totiž většinou autor článku, který by měl mít o problematice popisující v článku největší přehled. Výběr klíčových výrazů je však také hodně subjektivní záležitostí. Každý může výběr správných klíčových slov vnímat jinak. Někdo může jako klíčový termín zvolit výraz, který by někdo jiný jako klíčový vůbec nezvolil. Proto výrazy, které vysoko ohodnotil můj systém a nenacházejí se mezi klíčovými v článku, nemusí být ve všech případech špatným výběrem.

V této kapitole se pokusím analyzovat klíčové výrazy některého článku a porovnat je s výrazy, které vybral můj systém.

Článek, který jsem zvolil pochází z konference ASRU a jeho název je „*Speech enhancement using PCA and variance of the reconstruction error in distributed speech recognition*“. K článku je přiřazeno 6 klíčových výrazů:

- *speech enhancement*
- *speech recognition*
- *model identification*
- *signal subspace*
- *principal component analysis*
- *colored noise*

Nyní se pokusím vyextrahovat z tohoto článku klíčové výrazy pomoci mého systému. K tomu jsem použil morfologický analyzátor TreeTagger, stoplist, background vytvořený z článků konference a ohodnocovací metodu *tf-idf*, která v testech dopadla nejlépe.

Celkem bylo vybráno a ohodnoceno 826 kandidátních výrazů. Nejprve se podívám na ohodnocení klíčových výrazů, které k článku přiřadil autor. Výsledky najdete v tabulce 8.6. V prvním sloupci najdete pořadí výrazu v seřazené posloupnosti podle ohodnocení, ve druhém sloupci je klíčový výraz. Třetí sloupec obsahuje základní tvar tohoto výrazu.

pořadí	klíčové slovo	lemma
6.	*speech enhancement*	*speech enhancement*
818.	*speech recognition*	*speech recognition*
9.	*model identification*	*model identification*
1.	*signal subspace*	*signal subspace*
33.	*principal component analysis*	*principal component analysis*
-	*colored noise*	*color noise*

Tabulka 8.6: Tabulka klíčových výrazů.

V tabulce můžeme vidět, že klíčová slova byla ohodnocena relativně dobře. Hned 3 se umístily v první desítce. Klíčový výraz *speech recognition* byl ohodnocen velmi špatně. Důvodem zřejmě je, že se jedná o jakýsi obor kterým se článek zabývá. Tento výraz se pravděpodobně vyskytuje v textech mnoha podobných článků, ze kterých byl vytvořen background. Výraz *colored noise* nebyl ohodnocen vůbec, přestože se v článku vyskytuje. Důvodem je to, že základní tvar výrazu je podle TreeTaggeru *color noise*. Všechny výskyty výrazu v článku převedl TreeTagger právě na tento základní tvar, který byl v dalším zpracování ohodnocen. Výraz *color noise* byl podle ohodnocení na 207. místě.

V další části této kapitoly zkusím analyzovat 10 nejlepších výrazů vybraných mým systémem v pořadí podle jejich ohodnocení:

1. *signal subspace* – Tento výraz se mezi původními klíčovými slovy vyskytuje, takže je v tomto seznamu oprávněně.
2. *noise* – Toto slovo se v textu vyskytuje mnohokrát, většinou však ve spojení s jinými (*noise reduction*, *babble noise*, *colored noise*, *noise subspace*, atd.), takže bych ho přímo jako klíčové asi nevybral. Částečně však může být alternativou za výraz *colored noise*, který v tomto výběru není.

3. *subspace* – Taky se v textu vyskytuje mnohokrát ve spojení s jinými výrazy. Většinou však se *signal subspace*, což je hodnoceno ještě výše.
4. *reconstruction error* – Tento výraz je přímo v názvu článku, klidně může být klíčový.
5. *noisy signal* – Vyskytuje se v textu mnohokrát, klidně může být označeno jako klíčové slovo.
6. *speech enhancement* – Tento výraz se vyskytuje již mezi původními klíčovými slovy.
7. *distribute speech recognition* – Tento výraz je možná lepší, než výraz *speech recognition* vybraný autorem. Vyskytuje se v textu i v nadpisu, bohužel však v jiném tvaru.
8. *eigenvalue* – Jedná se o matematický pojem, asi bych to jako klíčové slovo nevolil.
9. *model identification* – Tento výraz se vyskytuje již mezi původními klíčovými slovy.
10. *wiener* – Toto slovo se v článku několik krát. Většinou samostatně, nebo ve výrazu *Wiener filter*. Asi bych ho jako klíčové nevybral.

V výsledku je patrné, že z desítky nejlépe ohodnocených výrazů se dá použít většina jako klíčová. Jak jsem však napsal výše, vše závisí na subjektivním pocitu autora a čtenáře článku.

Kapitola 9

Závěr

Hlavním cílem této diplomové práce bylo prozkoumat základní metody používající se k extrakci důležitých slov z článku a pokusit se porozumět charakteru používaných klíčových slov z dostupné množiny testovacích anglických článků. Na základě těchto zjištění se poté pokusit navrhnout a implementovat systém využívající tyto metody a nakonec zhodnotit dosažené výsledky.

V první části práce bylo potřeba nashromáždit co největší množství kvalitních testovacích dat. Celkem se mi podařilo stáhnout 11 074 článků s 39 440 klíčovými slovy. Tyto články většinou pochází z různých technických konferencí, najdeme však mezi nimi i články pocházející z jiných vědeckých oborů a technických časopisů. Některé konference měli k dispozici metadata, ve kterých byla klíčová slova jednotlivých článků identifikována. Ostatní články však měly klíčová slova uvedena přímo v textu. Bylo potřeba tyto slova vyextrahovat. To se podařilo. Články jsem pak sjednotil do jednoho XML souboru.

Nad testovacími články jsem poté provedl sadu různorodých testu, o nichž se rozepisuji v kapitole 6. Tyto experimenty mi do jisté míry objasnily typické složení v praxi používaných klíčových slov. Překvapilo mě, že zhruba 23% klíčových slov se v přiřazených článcích vůbec nenachází. Určitým řešením bylo použití morfologického analyzátoru, který převodem slov do základních tvarů tuto hodnotu snížil na polovinu.

V další části jsem se pokusil vytvořit výsledný systém. Systém se skládá ze 3 základních skriptů a je naprogramován v jazyce Python. Naimplementoval jsem metody výběru klíčových výrazů – *tf*, *tf-idf* a *ridf*. Ovládání skriptů je popsáno v kapitole 7.4.

Poslední část této práce jsem věnoval testování vytvořeného systému. Zjišťoval jsem, z jakého důvodu systém nenašel správná klíčová slova. Také jsem se pokusil zjistit, zda redukce kandidátních výrazů pomoci stoplistu a morfologického analyzátoru nějak omezí výběr správných klíčových slov. Zjistil jsem, že nejvýhodnější je použití morfologického analyzátoru. Přestože asi 3% klíčových slov odfiltruje, sníží počet kandidátních výrazů skoro na polovinu. V dalším testu jsem zjistil, že systém nejlépe ohodnocuje výrazy s použitím morfologického analyzátoru a metody *tf-idf*. V poslední části jsem se pokusil analyzovat výstupy mého systému.

Vytvořené skripty jsem testoval na školních serverech `merlin` a `athena[1-3]`. Možností dalšího vylepšení může být několik. Zajímavé by bylo vyzkoušet další statistické metody ohodnocení výrazů, které nebyly v textu zmíněny. Určitě by bylo dobré se zaměřit také na výběr kandidítních výrazů a pokusit se redukovat jejich počet.

Příloha A

Seznam zkratek TreeTaggeru

	značka	slovní druh
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	adjective, comparative
9.	JJS	adjective, superlative
10.	MD	Modal
11.	NN	Noun, singular or mass
12.	NNS	Noun, plular
13.	NP	Proper noun, singular
14.	NPS	Proper noun, plural
15.	PDT	Predeterminer
16.	PP	Personal pronoun
17.	RB	Adverb
18.	RBR	Adverb, comparative
19.	RBS	Adverb, superlative
20.	RP	Particle
21.	SYM	Symbol
22.	TO	*to*
23.	VB	Verb, base form
24.	VBD	Verb, past tense
25.	VBG	Verb, gerund or present participle
26.	VBN	Verb, past participle
27.	VBP	Verb, non-3rd person singular present
28.	VBZ	Verb, 3rd person singular present

Tabulka A.1: Přehled zkratek morfologického analyzatoru TreeTagger [8]

Příloha B

Nastavení proměnného prostředí

Z důvodu použití morfologického analyzátoru TreeTagger ve skriptu `parser.py` je ke správnému běhu tohoto skriptu potřeba nastavit proměnné prostředí. Toto nastavení platí pro školní servery `merlin` a `athena[1-3]`, na kterých jsem systém testoval:

```
export LD_LIBRARY_PATH=/mnt/minerva1/nlp/local64/lib:.
export LD_RUN_PATH=/mnt/minerva1/nlp/local64/lib:.
PATH=/mnt/minerva1/nlp/local64/bin:/mnt/minerva1/nlp/local/bin/:"$PATH"
export PYTHONPATH=/mnt/minerva1/nlp/local64/lib/python2.5
export MINIPATH=/mnt/minerva1/nlp/software/minipar/data
export PATH=/mnt/minerva1/nlp/software/TreeTagger/cmd/:$PATH
```

Literatura

[1] Oxford University Press: Oxford Journals. 2010, [Online; navštíveno 10. 4. 2010].
URL `http://www.oxfordjournals.org/`

[2] Proceedings of the Nat. Academy of Sciences. 2010, [Online; navštíveno 10. 4. 2010].
URL `http://www.pnas.org/`

[3] The Apache Software Foundation: *Apache Solr*. 2009, [Online; navštíveno 23. 4. 2010].
URL `http://lucene.apache.org/solr/`

[4] Dresto, E.: *Related docs*. FIT VUT v Brně, 2010, [Online; navštíveno 23. 4. 2010].
URL `https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Related_docs/`

[5] Knoth, P.; Schmidt, M.; Smrž, P.; aj.: Towards a Framework for Comparing Automatic Term Recognition Methods. In *Znalosti 2009*, 8th Annual Conference, Brno, Faculty of Informatics and Information Technology STU, 2009, ISBN 978-80-227-3015-0, s. 83–94.
URL `http://www.fit.vutbr.cz/research/view_pub.php?id=9065`

[6] Lokaj, T.: *GVP:Převod publikací do textového tvaru a jejich indexování*. FIT VUT v Brně, 2010, [Online; navštíveno 8. 4. 2010].
URL `https://merlin.fit.vutbr.cz/nlp-wiki/`

[7] Mašláňová, M.: *Automatická identifikace klíčových slov*. diplomová práce, Brno, FIT VUT v Brně, 2007.

[8] Santorini, B.: Part-of-Speech Tagging Guidelines for the Penn Treebank Project. 2010, [Online; navštíveno 18. 4. 2010].
URL `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/Penn-Treebank-Tagset.pdf`

[9] Schmid, H.: TreeTagger. 2010, [Online; navštíveno 8. 4. 2010].
URL `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

[10] Strachota, T.: *Automatická tvorba rejstříku publikace*. bakalářská práce, Brno, FIT VUT v Brně, 2008.

[11] Wikipedie: Python — Wikipedia.org: Otevřená encyklopedie. 2010, [Online; navštíveno 8. 4. 2010].
URL `http://cs.wikipedia.org/w/index.php?title=Python&oldid=5154116`

[12] Wikipedie: T test — Wikipedia.org: Otevřená encyklopedie. 2010, [Online; navštíveno 8. 5. 2010].
URL `http://cs.wikipedia.org/w/index.php?title=T_test&oldid=5311138`